$5
October 1986

ST Applications

THE ATARI 520/1040ST JOURNAL

IN THIS ISSUE:
Cover Contest Winners
Diskette Control
MIDI Reviews

10
0 74470 19763 8

InSoft, Corp. / InSoft **ST NETWORK**

For the low annual membership fee of $50, ST Network members can buy Hardware and Software for their ST's at Wholesale + 5%

********** SPECIAL TO NEW MEMBERS **********
With application for membership you can order software from the following publishers at 45% off list (normal discounts are 40% off list):
(add 5% to total, add $2 for each software package ordered for S/H)

Abacus	Grafikon	Priority Software
Absoft	Haba	Proco Products
Academy	Hisoft	Psygnosis
Access	Hippo	QMI
Accolade	Infocom	Quickview
Activision	Mark of the Unicorn	Regent
Adventure Inter.	Mark Williams Comp.	Royal
ANTIC	Metacompco	Sierra On Line
Artworx	Michtron	Shanner
Atari	Microprose	Soft Logik Corp.
(Exc. Devel. Pack)	Mindscape	Soft Technics
Batteries Included	Mirage	Softworks Limited
Bayview	Migraph	Spinnaker
Beckemeyer Dev.	Opt. Sys. Soft.	SSI
Better Working	OVS	SST
Central Point	Paperlogic	Synapse
DAC	Pecan	TDI
Dragon Group	Penquin	Unison World
Electronic Arts	Phylon	VersaSoft
Epyx	Power Series	VIP
Firebird	Prog. Comp. App.	Windham
FTL	Prospero	Xlent

ST HARDWARE SPECIALS
(list prices are in brackets, wholesale prices are first number)
(add 5% to total)

Abacus books - 25% off list
Atari Hardware
- 520 ST Keyboard ..... $340(500)
- SM 314 DS Disk Drive ... $180(299)
- Monochrome Monitor ..... $150(250)
- 20 MEG Hard Drive ..... $625(799)

Avatek 1200 Modem ..... $81(199)
Blank Discs - 3.5"
- BASF /10 SS DD ..... $14(29)
- Verbatim /10 SS DD ..... $14(29)
- Brown /10 DS DD ..... $20(39)
- Maxell /10 DS DD ..... $22(39)

Casio CZ101 Midi Syn. ..... $270(399)
IB 5.25" drives for the ST
- 40 Tracks ..... $190(270)
- 80 Tracks ..... $200(280)

Paradox 5.25" drives for the ST
- 40 Tracks ..... $190(270)
- 80 Tracks ..... $200(280)

Paradox IBM Software Simulator with 5.25" 40 track drive . $310(420)

EZ RAM (user installed) . $145(199)
(1 MEG upgrade, 13 solder points)
Hippo Eprom Burner ..... $85(139)
HippoVision Video Dig. ... $95(139)
Hippo Sound Digitizer .... $95(139)
Logichron Clock Card ..... $34(50)
Supra 20 MEG Hard Drive . $625(799)
Supra 1200ST Modem ..... $125(199)
ST to Par. Printer Cable .. $15(30)
ST to Disk Drive 6' Cable . $12(20)
QMI 1200ST Modem ..... $125(199)
Volksmodem VM520 ..... $130(199)
Supra MS 64k print buffer . $51(99)

InSoft's Magazine on a Disk
12 months—$70
6 months—$45

InSoft's C Tool Boxes
Math Tool Box—$59
Search/Sort Tool Box—$59
Graphic Tool Box—$59

InSoft ST NETWORK
MEMBERSHIP APPLICATION

P.O. Box 180
Boston, MA 02123

1-800-556-5580
1-617-739-9012

Compatibility of ST configuration with software/hardware is the responsibility of buyer

Restocking fee is 15% for merchandise not accepted

Products Ordered	Cost + 5%	S/H	Total
Membership—one year			$50
		Mass. residents add 5% sales tax	

☐ Check or Money Order enclosed for ..... $

Name _____
Address _____
City _____
Telephone _____ State _____ Zip _____

ST Applications
THE ATARI 520/1040ST JOURNAL
Vol. 2, No. 9 October 1986

Features:

4 You're a Pirate & I'm a Slick Operator
by Dick Biow

6 A Sampler of German Software-
by Maurice Kowalski & Bill Petry

22 Diskette Control-
by David Bessey

MIDI- conducted by Richard Kaller

12 MIDI Notes- News, etc.
14 MIDI Beat- Reviews of EZ Track &
ST Music Box

Departments:

18 FORTRAN as a Second Language / Mark Lovell
26 A Step FORTH: / Albert Lew
32 Not So, BASIC! / Eric Thornton
35 First Machine Language Program / Tom McAllister
38 You Can Do It! / David Bessey
40 Let's 'C' Now / David Stokes
43 The Joys of Modula-2/ST / Sol Guber
48 LOGO LOG / Leonard Kaplan
51 Beginner's Corner / Tom Hawkins

Reviews:

Review-Review

55 A Final Word in the Hand- Mark of the Unicorn

The ST Review Board

58 PECAN & Pro- Fortrans/
59 Leaderboard/ Access Software
61 Personal Money Manager/ Michtron
61 Multi Forth/ Creative Solutions Inc.
62 Mach2 Forth/ Palo Alto Shipping
64 Business Statistics/ Lionheart

69 ST User Groups
70 ST Bulletin Board Update
71 ST New Software/Hardware

Desk File

Editorial

Cover Contest Winners

Back in May we offered $100 to anyone who could come up with original artwork for our fall covers. Well, the response wasn't overwhelming. In all we received 6 entries from 5 people with a tie for first place. A flip of the coin chose this month's cover- Adrienne Badella's Checkmate. Next month our cover will display Marc Ritter's Expanding Atari. The winners will receive a check for $50, and our thanks. All other entrants will be offered a selected software package, and our thanks.

Disk Edition Available

Believe it or not ST Applications was originally intended to be a disk based magazine. It didn't take us long to realize that it would be difficult to share articles and programs on a disk in a car-pool ride to work. Thus was born ST Applications in its paper form. However, we have not abandoned our disk form. Every issue of ST Applications has been accompanied with a Support Disk. For the first four issues (Sept. to Dec. 1985) the disk version contained both the text files of all the articles and programs in the printed magazine topped up with PD programs etc. Starting with the Jan/Feb 1986 issue we changed the disk contents to be only the programs in the magazine. We then filled up the rest of the disk with PD programs etc., which were downloaded from various BBSs.

Volume 1 (1985), therefore, contained 4 separate disk issues. Volume 2 (1986), due to our Jan/Feb double month issue, will contain only 11 issues.

This month's disk is #9. Volume 3 will start with the January 1987 issue.

Pascal Editor

Chris Robertson has notified us that he will no longer be able to give us his full monthly support. So, we are looking for a new Pascal editor in residence to write monthly installments espousing the virtues of Pascal, patiently answer letters from readers who look for a helping hand to get them up & programming and accept only minimum payment- Please apply within.

Thanks for a good first year,

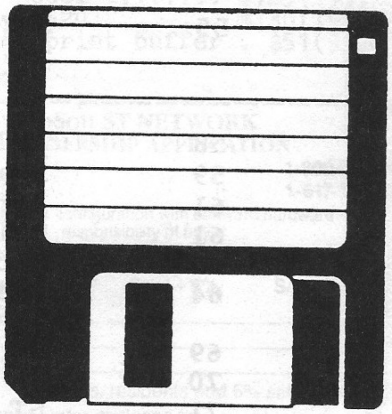
Bill Petry

Bill Petry
Publisher

Top Ten in Software Sales for August 1986

(These are preliminary figures from our monthly dealer sales survey. Final figures will appear in our Monthly Dealer Survey Results, which is mailed free to participating dealers.)

1. **Silent Service-** *Microprose*
2. **Leaderboard-** *Access Software*
3. **Thunder-** *Batteries Included*
4. **Time Bandit-** *Michtron*
5. **Winter Games-** *Activision*
6. **Flash-** *Antic Software*
7. **Mean 18-** *Accolade*
8. **Music Studio-** *Activision*
9. **Sundog-** *FTL Games*
10. **Joust-** *Atari*



ST Applications

THE ATARI 520/1040ST JOURNAL

Vol. 2, No. 9 October 1986

Editor and Publisher
William Petry

Assistant Editors
Richard Kaller
Sharon Richmond

Contributing Writers:
Sol Guber, Leonard Kaplan, Albert Lew, Mark Lovell, Tom McAllister, David Stokes, and Eric Thornton

Technical Consultant:
Brett Morgan

ST Applications
President
William Petry

ST Applications- The Atari 520/1040 ST Journal (ISSN 0887-3755), is copyright © 1986, and published monthly by ST Applications, P.O. Box 980, 10760 Hwy 116, Forestville, CA 95436.

North American subscriptions are $40 per year and payable by Check or Money Order in United States currency only. Foreign Air Mail subscriptions are $78 per year. For subscriber and advertising service please call (707) 869-2325.

ST Applications is in no way affiliated with Atari Corporation.

Atari, 520ST, 1040ST and the Atari logo are registered trademarks of Atari Corp.

GEM is a registered trademark of Digital Research Inc.

To the best of our knowledge the information contained herein is correct. However, ST Applications, it's editors, and staff cannot be held responsible for the use of the information or any damages which may result.

Application to Mail at Second-Class Postage Rates is Pending at Forestville, CA 95436.

Postmaster: Send address changes to ST Applications, P.O. Box 980, Forestville, CA 95436.

Bigger is better...

Improve your 520ST

with a ThoughtSpace one meg memory expansion board

The TS-1A gives you greater speed and power, bigger documents and applications, more desk accessories, larger RAM disks, and full 1040ST software compatibility. There are no special programs to run or changes to the system software.

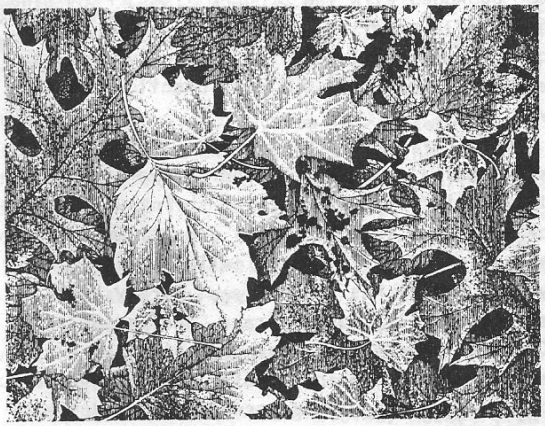
Have it installed or do it yourself (it's not hard with our detailed, illustrated instructions). You always get a complete, tested board with a utilities diskette and a **one year guarantee.**

Contact your Atari dealer for information or write or call ThoughtSpace for literature. Dealer inquiries invited.

ThoughtSpace Development

2450 Warring St., No. 21 Berkeley, CA 94704 (415) 845-1415

COMDEX®/Fall '86



November 10-14, 1986

Las Vegas, Nevada

by Dick Biow

You're a Pirate and I'm a Slick Operator

This article represents the views of Dick Biow and does not necessarily represent the views or opinions of ST Applications or its staff.

The word, "piracy," can be taken to mean many things but, in the business world, it means, "any predatory practice." In the light of that definition, the most flagrant examples of piracy in the field of computer software can be discovered among the software publishers, themselves.

Consider, for example, those publishers who claim fantastic performance in their promotional literature and then deny all resposibility for performance in the shrink-wrap disclaimers that you discover packaged with the program disks. Promising you benefits in order to get your money and then retracting those promises after getting your money is clearly a "predatory" business practice: it is piracy.

But aren't shrink-wrap disclaimers legal? In many states, especially where publishers have lobbied legislatures into compliance, the shrink-wrap gimmick stands up in court. That doesn't make it ethical; it still fulfills the definition of piracy, but now it has become **legal** piracy. (The British pirate, Sir Francis Drake, carried a letter from Queen Elizabeth I of England, authorizing him to board and capture Spanish vessels. Drake was a legalpirate, but his victims probably missed the subtlety.)

If publishers practice piracy, what about their customers? What about consumers who back up program disks without a publisher's permission? What about people who not only make unauthorized backups, but then proceed to share them? One publisher's association, ADAPSO, is trying to popularize the idea that illicit copying is simple thievery, like shoplifting. Few consumers accept that viewpoint or even take it seriously. Why? Why do the moral pronouncements of an organization representing publishers have such low credibility?

One reason, of course, is the arrogance of such preaching when contrasted with popular marketing practices. Consider, for instance, the common use of copy-protection by many publishers: the manipulation of software formats so as to physically prevent backups. Hypocrisy here is too evident to permit credibility: at one and the same time, these copy-protectors are (1)appealing to your nobler instincts and (2)treating you like a crook by forcibly trying to block you from doing the very thing they ask you not to do as a matter of principle!

Are the folks who can produce such fine programs really so illogical? That's unlikely. I believe they have been psyched by publishers of copy-protection **programs** into a sort of anti-consumer hysteria. It has to be hysteria--it can't be rational--because copy-protection doesn't do a thing to defeat piracy.

I personally have purchased five different programs designed to defeat copy-protection schemes. Any three of them will back up everything now available for the ST. All copy-protection does is enrage consumers, who find they can't transfer programs to double-sided or hard disks. And that

invites retaliation--"do 'em as they do you"--in the form of **extremely wide** illicit distribution.

The sharing of programs without a publisher's permission is illegal, no question. But the question is, **should** it be? Clearly, most consumers do "share" to a moderate degree. They tend to look at the practice the way wine -drinking must have been looked at during Prohibition--as a private right that no law may inhibit, and where the judgment of what may be ethical must always be made by the individual. (Political science scholars like to point out that laws never impede activities that society considers permissible.)

No, I am not saying that the irresponsible distribution of programs is proper, and I certainly don't believe anyone has the moral right to sell someone else's copyrighted work. What I do object to is the suggestion by industry spokesmen that every undocumented copy of a program represents a lost sale to a publisher. That is like proposing that everyone who ever got a pass to Disneyland would have bought a ticket otherwise.

Suppose two consumers cannot individually afford an expensive program they both admire. So they combine forces, each paying half, and one keeps the original disk while the other accepts a copy. Again, that undocumented copy is illegal but again, should it be? Who get hurt? Nobody. Who benefits? Everybody. ADAPSO will tell you that the illicit copy has robbed a publisher of a sale, but clearly that illicit copy is what made it possible for the publisher to **make** the sale. To think otherwise is reasoning from greed--not logic.

How can publishers persuade the buying public to cooperate, to limit undocumented copying? Primarily, I think, they must stop treating consumers like adversaries. This means the termination of preachy and surly attacks on consmumers by industry spokesmen. It means an industry-wide abandonment of copy-protection. It means a policy of fair and just shrink-wrap agreements: reasonable guarantees instead of disclaimers. Overall, it means setting a standard for the type of behavior that publishers hope to elicit from their customers.

Perhaps my own attitude toward **FinalWord** by Mark of the Unicorn is illustrative. **FinalWord** is my working tool as a writer. I use it daily; I am using it now. And no, I have never given away a copy of it, though I can defeat its copy-protection easily. I don't give copies away because I don't think that would be right; I think the program is fairly proced and honestly advertised. I even feel beholden to the publisher for the prompt and professional telephone support its staff always provides. But notice--this is **my** decision. Somebody **earned** my loyalty. Nobody intimidated me through preachy literature, ramrodded laws, nor packaged threats, and I don't think anyone ever will.

Simply put: if publishers will respect the attitudes of consumers, consumers are likely to respect the attitudes of publishers. Nothing else will work: Publishers must eventually do unto consumers as they would have consumers do unto them.

Submitting Material

Contributions to ST Applications are welcome from everyone. We want a variety of programs which can be helpful, useful or just fun for other Atari 520/1040ST owners.

We prefer articles with accompanying programs (and, hopefully, at least one picture or graphic) which demonstrates a type of programming style or application. We wish to keep the content of articles serious enough to enable the readers to refer back to them in the future- something to build on. This makes for a more meaningful relationship with the user & his/her computer. Chances are if a topic interests you, other people will find it useful as well. Our need for short articles is never ending.

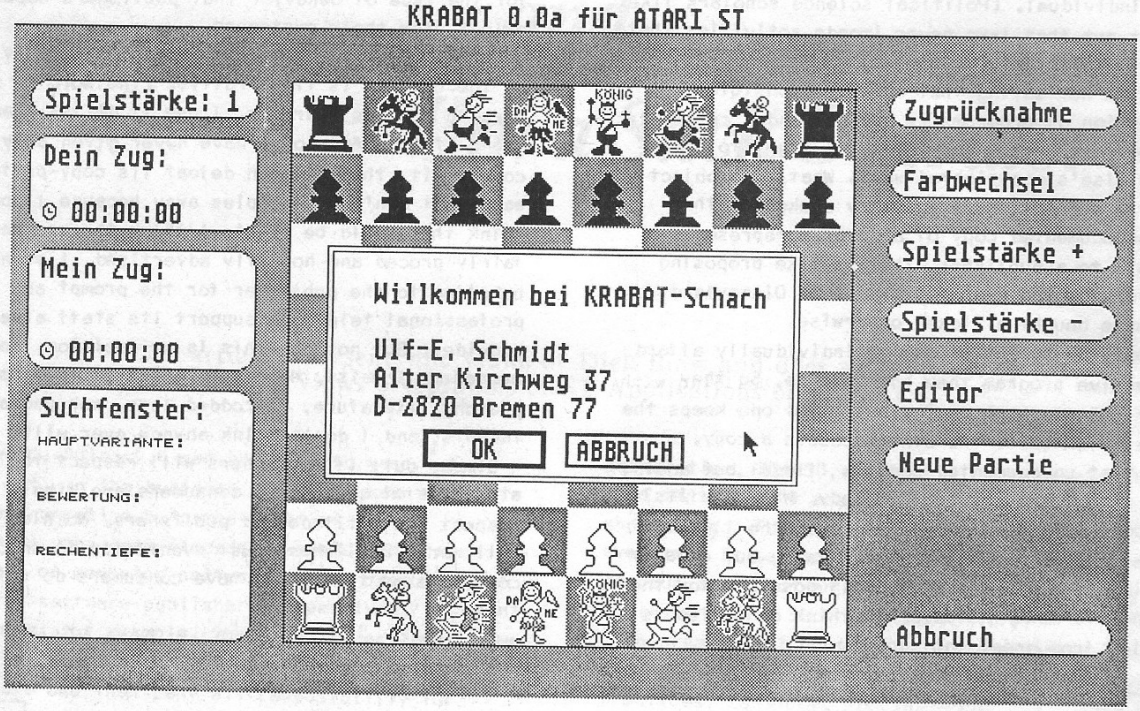
Should your submission include a program, please send in on disk if it is over 10 lines long. We just don't have time to type in and debug any programs longer than that. You are invited to include the text of your accompanying article on the disk as well as any screen art you may have. However, please send a hardcopy of the article, pictures and program listing. Please indicate which word processor you used.

Do us the courtesy of not submitting programs or articles currently submitted to another publication.

We do pay for submitted articles. However, we will normally offer subscriptions (or extensions) for your work. Those wishing remuneration should state that when making submissions.

Our Address: ST Applications, P.O. Box 980, Forestville, CA 95436

A Sampler of German Software



by Maurice Kowalski & Bill Petry

Last spring I chanced to meet Maurice Kowalski, a visiting student from Berlin, at the West Coast Computer Faire in San Francisco, CA. This month Maurice shares with us his views on a German public domain chess game and the 'Pearl' multi-tasking operating system developed for the ST.

We also give a few screens of other German PD and commercial programs.

Should you be interested in other sources of German and German-language software for the ST we give the following contacts which we have found of value.

Atari "Germany" Corp.
Frankfurterstrasse 89-91
6096 Raunheim
West Germany

These three magazines are quite interesting for Atari ST information and software in West Germany.

Atari ST Computer (strictly Atari ST)
(93DM for foreign Airmail)
Heim-Verlag
Heidelberger Landstrasse 194
6100 Darmstadt-Eberstadt
West Germany

level 16 (for 16 bit computers incl. Amiga, MAC & ST)
(33DM for 6 issues)
Otto-Hahn-Strasse 26
4000 Dusseldorf
West Germany

c't (covers many computers, but does have Atari ST specific articles and projects)
(104DM for 12 issues)
Verlag Heinz Heise GmbH
Bissendorfer Strasse 8
Postfach 61 04 07
3000 Hannover 61
West Germany

KRABAT/CHESS
Reviewed by Maurice Kowalski

The history of chess began centuries ago and with it, the first chess problems. In modern times the problem was: is it possible that a machine (computer) can play chess? Later in MIT labs the first simulation of a chess program was written. Two professors were asked to play against a new chess program. The programmers were told they were playing a computer, when in fact they were playing each other. One professor was in one building with a terminal and the other in another building with another terminal. Neither professor knew the other before starting to play. Actually, they recognized the fake 'program' because they noticed the moves came randomly without special time limits.

Later on real programs were designed using chess as a topic, and the chess software started improving in speed, skill, graphics, etc. up to today.

What I'm presenting today is one of the better chess programs for the ST- **KRABAT**-A German chess program.

After I went to a friend with an Atari ST and a monochrome monitor (necessary-high resolution) and started the program, a very well organized menu and a chessboard appeared in front of my eyes. After a couple of mouse 'clicks' my game started. The program is quite strong and skillful, but it doesn't solve the problem of self sacrifice totally. That is still a deficiency where you can catch a chess program. However, I think it will satisfy beginners up to intermmediate players.

The second part of the program is a chess editor (the best mouse driven I have ever seen). You 'click' one field and then you 'click' until you have the right figure at the field. One thing that bothers me is that you can't enter moves by notation.

Another part of the program is that you can change the look of the figures by an icon-editor. The program has all the features that a chess program has to have (clock, move back, skill change, color change, etc.)

The program is incredible for the price. It is public domain and for about $12-$13 you get the latest version when ordering from the author.

Above all, it is completely mouse driven and the program has all of the important features excluding player vs. player and en passant moves (you have to do en passant by the editor). I would say this program is a 'must' in any software library. The only thing that you need is a German-English dictionary.

For information write to:

E. SCHMIDT
Alter Kirchweg 37
2820 Bremen 77
West-Germany

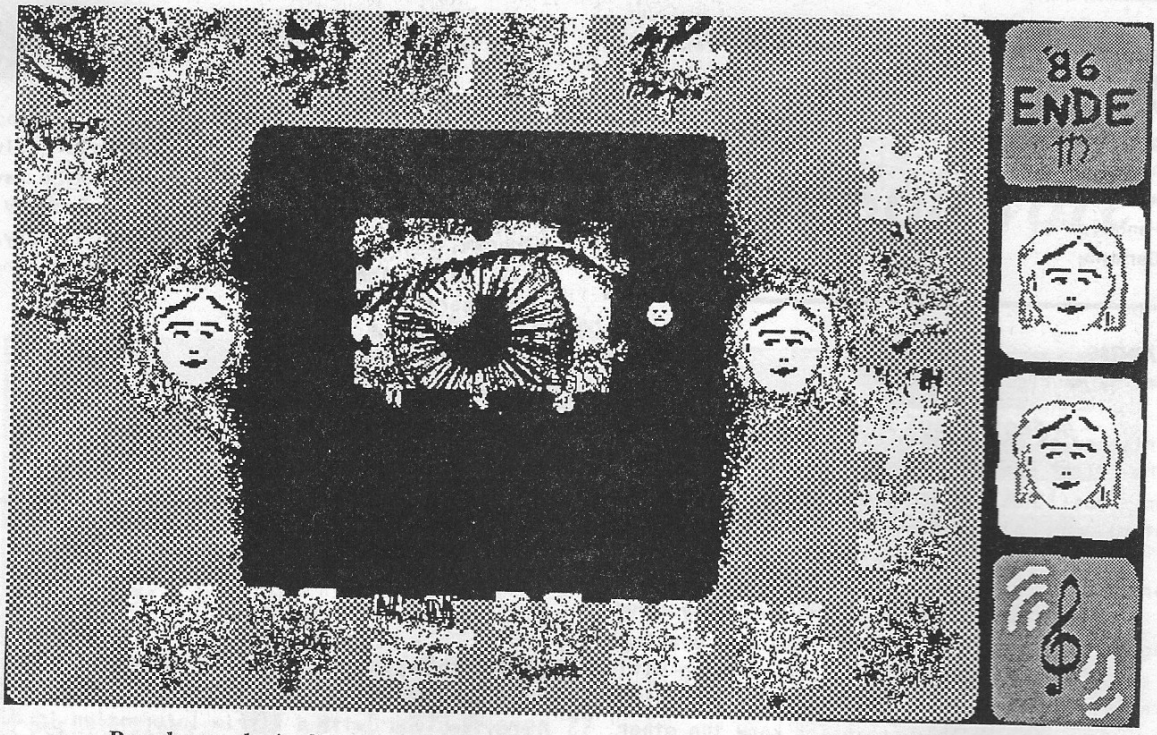
RTOS-UH/PEARL
by Maurice Kowalski

In this article, I will try to introduce to you the operation system of the **ROTS-UH/PEARL**. Let me provide you with a little information in advance. RTOSstands for Real Time Operation System. This system was programmed at the University of Hannover in West Germany (UH). It was also transferred to the Atari ST compputer in Hannover. Let me explain a few things about PEARL. This system actually is a compiler which is implememted for the M68000.

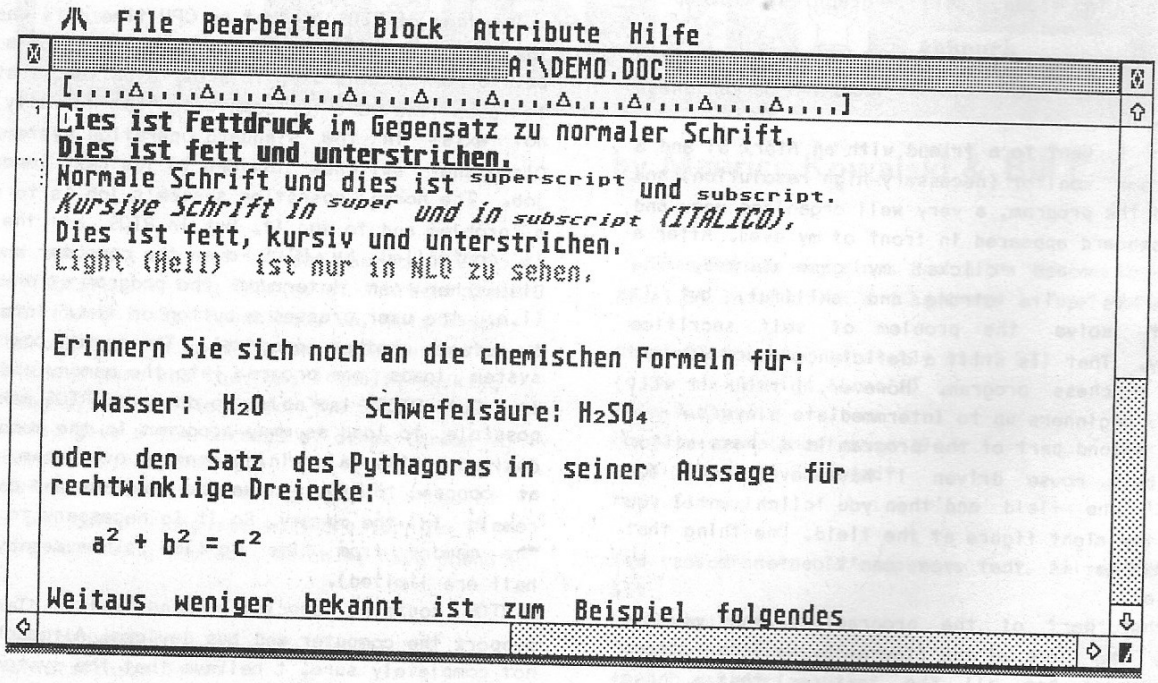
The idea of RTOS is that no CPU time gets wasted. The computer no longer has to wait for a slow printer or for the user to press a button. Instead, it uses the CPU for something that normally does not exist in the standard operation system. The Dispatcher switches the CPU to the next important job. The normal operation system's job is to start a problem and to end it. But in RTOS, all the time is controlled by RTOS over the computer and the Dispatcher can interrupt the program at any time (i.e., the user presses a button or timer interrupt to start another program). The normal operation system loads one program into the memory and runs it. But RTOS is able to do more. RTOS makes it possible to load as many programs in the memory as desired while maintaining control over them. This, as opposed to TOS, allows the programs and data to remain in the memory. So it is necessary to clean the memory from time to time (since megabyte or half are limited).

RTOS contains special commands called "Task" to support the computer and bus devices. Although I am not completely sure, I believe that the system is a bi-user system. I was only able to access some, but not all of the information. RTOS also has a couple

Continued on page 11 -->



Puzzlepuzzle is Shareware by Tommysoftware, Mainzer Landstr. 147 6000 Frankfurt / M. 1, West Germany. If you like it they request you send $10. If you'd like the Pascal source (in CCD's Pascal 1.4), send an additional $10.



This is a screen from ATARI-Software GST's ATARI-1st Word, Deutsch version. ATARI...wir machen Spitzentechnologie preiswert.

STplus™

P.O. 1197 Berkeley, Ca. 94701

THIS MONTH'S SPECIAL: TEAK DISK HOLDERS: 50 disk roll top case for $22.50 , 100 disk case for $34.00. A nice accessory for your ST. Sony diskettes available, 10 for $16, double sided, 10 for $20. Supra 60 meg hard disk: $1700 VERY FAST DISK ACCESS!

BUSINESS

DBMAN	$104.95
EZ – CALC	$48.95
A – CALC	$41.95
SBT ACCOUNTING	(VARIOUS)
VIP PROFESSIONAL	$125.95
DAC – EASY ACCTING	$42.00 ST
ONE – WRITE	$69.95
REGENT BASE(GEM)	$79.79
ST ACCOUNTS	$166.56

GRAPHICS

DEGAS ELITE	$55.95
EZ DRAW	$104.95
GRAPHIC ARTIST	$142.95
PAINTPRO	$35.00
PC DESIGN BOARD	$276.50
B & W HIPPO DIGITIZER	$111.95

HARDWARE

1040 ST COLOR	$949.20
MONOCHROME	$799.20
520 ST COLOR	$779.20
MONOCHROME	$639.20
SUPRA or ATARI 20MB	$639.20
BROTHER 1109 MATRIX	$239.20
ATARI SMM804 MATRIX	$183.20
RICOH 2200, 22CPS LQ	$387.95

ENTERTAINMENT(CALL FOR MORE!)

WINTER GAMES	$27.00
PHANTASIE	$27.00
DIABLO	$21.00
UNIVERSE TWO	$48.95
LEADERBOARD	$27.00
STAR – RAIDERS	$21.00
JOUST	$ 21.00
SUNDOG	$27.00
SILENT SERVICE	$27.00

PROGRAMMING & UTILITIES

DISK DOCTOR	$21.00
FORTRAN 77	$104.95
PERSONAL PASCAL	$52.45
DISK LIBRARY	$34.95
MARK WILLIAMS 'C' &ASSM.	$125.95
MEGAMAX "C"	$179.00

WORD PROCESSING &TOOLS

COLOR WRITER	$48.95
HIPPO WORD	$62.95
THUNDER(spelling checker)	$27.95
FINAL WORD	$104.95

TELECOMMUNICATIONS

IS TALK	$34.95
FLASH	$27.95
ANSI GRAPH (VT102)	$55.97

We are totally commited to the ST line of computers. This ad is two months old, so if you don't see it, call us. We will carry all software for the ST at 30 % off and related hardware for 20% off. For consistently fast, competent, support and service call:

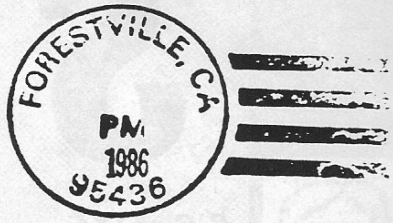
CALIFORNIA (800) 874 – 4789 WE SHIP ANYWHERE!

NATIONAL (800) 433 – 6222

LOCAL (415) 849 – 8717

VISA MC AMEX

Letters to ST



Dear ST,

I am the president of the local ST users group STAC-K (ST Atari Computers- Korea) and was wondering what, if any, discount you give to user groups towards large quantity subscriptions. We feel that in order to use the programs in your magazine that each person should have a copy, so as not to infringe on the copyright. Please inform us so that we can place our order. Also please keep producing the great magazine that you have started.

Just some other notes to pass on to your readers. ST*SIG disk magazine seems to have bitten the dust. I sent Clay Gradis a money order for issues 4-8 and nothing. He cashed the money order but didn't produce the disks. Appears to be another flash in the pan ripoff of ST owners. The same appears to be true for ASTUN in Utah, a couple of issues and then nothing. The new Compute! ST magazine is good but you can see that they are still tied to the corporate strings of advertisers.

Some of the software houses are sending out sample programs to the user groups which we here in Korea are really happy about. It gives us a chance to check out the program before we shell out the money. We hope that other houses will be doing this in the future. Some of the members have already ordered the complete programs based on what they saw in the demos.

-Robert Mann
APO SF 96224-0373

Robert,

We currently offer user group subscriptions for $30 each when paid for in batches of 5 or more at a time. This offer includes one disk subscription per user group. Our policy on program use and possession is that we reserve the right to sell these programs in ST Applications and our companion support disk. User groups have our permission to offer the programs on our monthly disks to other members of the group free of charge thru their group BBS.

-Bill Petry

Dear ST,

For those keeping track of ST BASIC bugs, this program illustrates another one. ST BASIC allows you to have strings of up to 255 characters. However, you evidently cannot get to parts of a long string beyond character 127. When this program is run, you will see that the MID$ function returns a string of 10 "A"'s, as does the RIGHT$. But the string AB$ contains all "B"'s from position 128. If you change line 40 to ?MID$(AB$,127,10) you get one "A" and 9 "B"'s. With this in mind I would recommend avoiding the use of strings longer than 127 characters.

```
10 A$=STRING$(127,"A")
20 B$=STRING$(128,"B")
30 AB$=A$+B$:?AB$
40 ?MID$(AB$,128,10)
50 ?RIGHT$(AB$,10)
```

-Donald Krell
Forestville, CA

Errors & Boo Boos

In Ron Schaefer's DISPLAY program the following line was left out from the listing on page 15 of the August issue:

```
4650  b = 0:gotoxy 25,0:?"Touch NO to quit"
```

The NO is in Reverse (inverse).

In Leonard Kaplan's LOGO LOG in the September issue (pages 51 & 52) the three referenced LOGO listings were not included with the article. They are as follows:

```
TO PIGLATIN :XX
MAKE "MAXDO COUNT :XX
MAKE "COUNTER 1
;*************************
LABEL "AGAIN
;*************************
MAKE "CURRENT ITEM :COUNTER :XX
IF (MEMBERP FIRST :CURRENT [A E I O U Y]) [MAKE "CURRENT LPUT "Y :CURRENT]
TYPE BUTFIRST :CURRENT
TYPE FIRST :CURRENT
TYPE "AY
TYPE CHAR 13
TYPE CHAR 10
MAKE "COUNTER :COUNTER + 1
IF (:COUNTER <= :MAXDO) [GO "AGAIN]
END
```

Listing 1

```
TO FACTORIAL :N
IF :N = 1 [OUTPUT 1] [OUTPUT (PRODUCT :N FACTORIAL :N - 1)]
END
```

Listing 2

```
TO PIG2 :XX
IF (COUNT :XX > 1) [PIG2 BUTLAST :XX]
MAKE "CURRENT LAST :XX
IF (MEMBERP FIRST :CURRENT [A E I O U Y]) [MAKE "CURRENT LPUT "Y :CURRENT]
TYPE BUTFIRST :CURRENT
TYPE FIRST :CURRENT
TYPE "AY
TYPE CHAR 13
TYPE CHAR 10
END
```

Listing 3

More German Software from page 7

of other tasks, but I won't explain these here because these explanations will take up too much space. But it is worth mentioning that RTOS has a command for formating disks which is quite incredible. It is possible to get 83 or more tracks that all depend no longer on the software, but on the quality of the disk drive. RTOS also has an editor which is not fancy, but is quite adequate for writing source-files.

There are a few things that I didn't mention in the article. RTOS works with keyboard and command line. Consequently, no mouse or pull-down menus can be used (the commands are sometimes pretty long). Thus, I would recommend this system for programmers and not for users. Next, RTOS disks don't run on TOS, but it is possible to change RTOS disks from one computer to another. You have to think of RTOS like CPM or MS-DOS. RTOS also needs a c't userboard to run. If someone is interested enough in the c't userboard, I will be able to send the plans for the userboard. Please note that the RTOS-UH/PEARL costs around $80-$100.

MIDI Notes

by Richard Kaller

Things just keep looking better and better for the ST, particularly in the field of music and MIDI. Talk in the industry has it that the ST stands the best chance of becoming the 'Commodore 64' for MIDI musicians interested in 68000-based computers. Lest any Atari fanatics take this observation wrong, let me say it is meant as a compliment. Musicians, except for perhaps rock stars and the independently wealthy, are, just like everybody else, subject to the reality of having a limit to the amount of money they can spend on equipment, being required to keep some funds around for such trivial pursuits as food, shelter, etc. Keyboard players are especially affected by this because of the high price of keyboards, though this is changing somewhat as the latest technology provides decent sounding boards at a fraction of the cost several years ago. Still, as the C-64 was the least expensive option for getting a foot in MIDI's door with the widest array of software, many musicians chose it as their first computer. Now, the ST finds itself in a similar price position, and MIDI software companies, discovering that Atari is not going to dry up and blow away, are providing the support necessary to further solidify the ST's niche. This usually proves to be self-reinforcing. As more MIDI software is written for the ST, more musicians, and others, will buy one; as more people buy STs, the software houses are motivated to develop more programs.

The result for me and you, right now at least, is, not one, but two software reviews this month. Hybrid Arts has released **EZ Tracks**, a musician-oriented sequencer. (By the way, their CZ-Droid, a patch editor/librarian/creator is in the mail. Stay tuned for further details).... **ST Music Box**, from XLENT Software, is a composing tool designed to enable beginners to do from the ST what they may not yet be able to do directly with a musical instrument. XLENT has plans to use this as a base module for a line of MIDI programs... I know I continue to promise some reviews of keyboards and other MIDI hardware, but as mentioned above, I, too, have to control my urges to buy every new keyboard that comes along, and even when I can gain access to these boards, I do not always have time to become familiar enough with them to give a clear, concise, and exact picture of them. So, anybody out there who may have bought a new synth (that is, a model that is a year or less old) who is willing to spend an hour or two writing (I really mean word-processing; use **ST Writer** or **1st Word** please) about its strengths and weaknesses should do so and send the results to me, c/o **ST Applications.** Who knows; your work could grace these pages, with posterity ever-thankful (we would be, too, and we do pay for reviews). The main requirement is that your review remains coherent.... For those of you who do not consider yourselves budding Hemingways, but are still interested in seeing some equipment reviews, send your checks to the 'I'm So Broke I Can't Pay

Attention Fund'. I guarentee to spend all money recieved buying new keyboards. For review purposes, of course.

Speaking of reviews, I must say I've gotten some flak recently concerning mine. The main complaint seems to be that I only discussed what a piece of software can do and somehow left out what it can't. Though this point of view is understandable, there are a couple of things I'd like to bring up. First, these reviews are meant mainly to provide more details than contained in product releases. They are not meant, in most cases, to encourage or discourage sales, but to give you enough information to decide where to start. If the review describes abilities that sound like the program might fit your needs, go to a dealer or an acquaintance who has the program in question and have it demonstrated to your satisfaction. If you have a specific need for certain extra capabilities, make sure that the program will do those things in conjunction with the equipment you currently own before buying it. Second, as most of the MIDI products so far reviewed are first versions, often released to gain a foothold in the market, before the product has been upgraded to its maximum powers, it is difficult to start listing the things I wish they would do; a list like this could run as long as the basic review. Where I felt a piece of software was extremely unworkable or missing important functions I have tried to mention it. Third, a piece of software cannot be all things to all people. I have been playing music for over twenty-five years and closely following MIDI since its public appearance four years ago (in fact, MIDI is the main original reason for my involvement with computers). Most of the ST MIDI software thus far hasn't begun to meet my wants and desires, but that doesn't mean that someone else with an ST may not find this same software totally sufficient for theirs. For example, while I never use Q-R-S's MIDI Magic to play computer Muzak, others may be ecstatic about being able to do so. 'Nuff said.

Those programmers who are willing to pick up the gauntlet, so to speak, and write their own entries for the MIDI market but still need a little help talking to the ST MIDI ports may want to check out Abacus's tenth volume in the ST Reference Library, **Introduction to MIDI Programming for the Atari ST**, by Len Dorfman and Dennis Young (from XLENT Software), to be released in the end of September. We haven't gotten a hold of a copy yet for review, but it is supposed to include source listings for a "...comprehensive MIDI editor, driver, and animated player...".

Coming next month will be a review of Dr.T's **CZ Patch**, a product for which everyone with a Casio CZ

Continued on page 17 -->

ATTN: PASCAL USERS

MODULA-2
the successor to Pascal

FOR ATARI 520ST

- FULL interface to GEM DOS, AES and VDI
- Smart linker for greatly reduced code size
- Full Screen Editor linked to compiler locates and identifies all errors.
- True native code implementation (Not UCSD p-Code or M-code)
- Sophisticated multi-pass compiler allows forward references and code optimization
- Desktop automates Edit/Compile/Link cycle
- FileSystem, RealInOut, LongInOut, InOut, Strings, Storage, Terminal
- Streams, MathLib0 and all standard modules
- Directory search paths
- Supports real numbers and transcendental functions ie. sin, cos, tan, arctan, exp, ln, log, power, sqrt
- 3d graphics and multi-tasking demos
- CODE statement for assembly code
- 370-page manual
- Installs on Hard disk and RAM disk
- No royalties or copy protection
- Phone and network customer support provided

Pascal and Modula-2 source code are nearly identical. Modula-2 should be thought of as an enhanced superset of Pascal. Professor Niklaus Wirth (the creator of Pascal) designed Modula-2 to replace Pascal.

Added features of Modula-2 not found in Pascal

- CASE has an ELSE and may contain subranges
- Programs may be broken up into Modules for separate compilation
- Machine level interface
  - Bit-wise operators
  - Direct port and Memory access
  - Absolute addressing
  - Interrupt structure
- Dynamic strings that may be any size
- Multi-tasking is supported
- Procedure variables
- Module version control
- Programmer definable scope of objects
- Open array parameters (VAR r: ARRAY OF REALS;)
- Elegant type transfer functions

Ramdisk Benchmarks (secs)	Compile	Link	Execute	Optomized Size
Sieve of Eratosthenes:	6.2	4.3	3.5	2600 bytes
Float	6.4	4.8	8.3	4844 bytes
Calc	5.5	4.2	3.3	2878 bytes
Null program	5.1	3.2	—	2370 bytes

```
MODULE  Sieve;
CONST   Size = 8190;
TYPE    FlagRange = [0..Size];
        FlagSet = SET OF FlagRange;
VAR     Flags: FlagSet;
        i:     FlagRange;
        Prime, k, Count, Iter: CARDINAL;
BEGIN   (*$S-,$R-,$A+ *)
  FOR Iter:= 1 TO 10 DO
    Count:= 0;
    Flags:= FlagSet(); (* empty set *)
    FOR i:= 0 TO Size DO
      IF (i IN Flags) THEN
        Prime:= (i * 2) + 3; k:= i + Prime;
        WHILE k <= Size DO
          INCL (Flags, k);
          k:= k + Prime;
        END;
        Count:= Count + 1;
      END;
    END;
  END;
END Sieve.
```

```
MODULE Float;
FROM MathLib0 IMPORT sin, ln, exp,
                     sqrt, arctan;
VAR x,y: REAL; i: CARDINAL;
BEGIN (*$T-,$A-,$S-*)
  x:= 1.0;
  FOR i:= 1 TO 1000 DO
    y:= sin (x); y:= ln (x); y:= exp (x);
    y:= sqrt (x); y:= arctan (x);
    x:= x + 0.01;
  END;
END float.
```

```
MODULE calc;
VAR a,b,c: REAL; n, i: CARDINAL;
BEGIN (*$T-,$A-,$S-*)
  n:= 5000;
  a:= 2.71828; b:= 3.14159; c:= 1.0;
  FOR i:= 1 TO n DO
    c:= c*a; c:= c*b; c:= c/a; c:= c/b;
  END;
END calc.
```

Product History

The TDI Modula-2 compiler has been running on the Pinnacle supermicro (Aug. '84), Amiga (Jan. '86) and will soon appear on the Macintosh and UNIX in the 4th Qtr. '86.

Regular Version $79.95 Developer's Version $149.95 Commercial Version $299.95

The regular version contains all the features listed above. The developer's version supplies an extra diskette containing a symbol file decoder – link and load file disassemblers – a source file cross referencer – symbolic debugger – high level Windows library Module – Ramdisk and Print Spooler source files – Resource Compiler. The commercial version contains all of the Atari module source files.

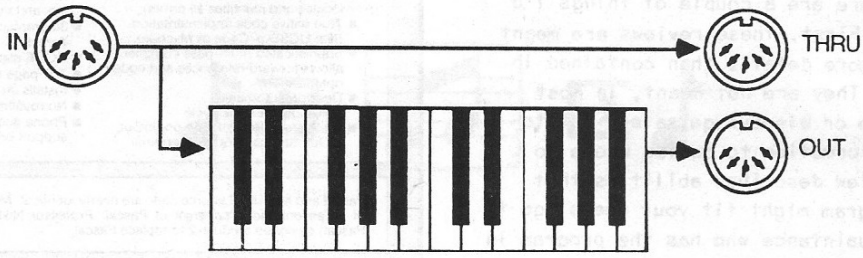
Other Modula-2 Products

Kermit	– Contains full source plus $15 connect time to Compuserve.	$29.95
Examples	– Many Modula-2 example programs to show advanced programming techniques	$24.95
GRID	– Sophisticated multi-key file access method with over 30 procedures to access variable length records.	$49.95

TDI SOFTWARE, INC.

10410 Markison Road ■ Dallas, Texas 75238 ■ (214) 340-4942
Telex: 888442 Compuserve Number: 75026,1331

MIDI Beat



by Richard Kaller

```
*********************************************
*                                           *
*                  EZ Track                 *
*                                           *
* Type: Sequencer                           *
* Level: Beginner/Intermediate              *
* Requirements: MIDI synth                  *
* Features: 20 tracks, overdubbing          *
* List price: $65.00                        *
* Publisher: Hybrid Arts, Inc.              *
*            11920 W. Olympic Blvd.         *
*            Los Angeles, CA 90064          *
*            Office - (213) 826-3777        *
*            BBS - (213) 826-42848          *
*                                           *
*********************************************
```

At last, a multi-track sequencer. One of the things that has intrigued me the most about MIDI from the beginning was the ability to record music (albeit in a symbolic, rather than analog, manner), in multi-track. The similarities to a professional tape recording studio are great, while the amount of extra control gained from digital manipulation of information is powerful, indeed. **EZ-Track** gives a musician the chance to bring multi-tracking home.

EZ-Track is a GEM-based program complete with pull-down menus and windows, controllable from the ST keyboard, or the mouse in most cases. It runs in medium and high resolutions and, of course, requires at least one MIDI synth or drum machine; if you are planning on doing any multi-track recording that makes use of more than one MIDI channel, you may need more instruments. With 20 tracks available, you can lay down parts from your synth, one track at a time, and sync them all together simply. Total note capacity is around 27,000 notes on a 520 and 63,000 on a 1040, large enough for most purposes. I say 'around' because **EZ-Track** is capable of recording all MIDI data, including velocity, pitch and modulation wheels, and program changes. There is a MIDI Clock output for driving drum machines. A MIDI Thru function is provided with a channelizing option. One note: the program won't run on the early 520s with TOS still loaded from the disk; check with your local computer dealer about getting TOS in ROM.

The **EZ-Track** screen is divided into two parts, the Control Display on the upper right and the Track Display on the left. These function somewhat like a tape recorder's controls and a mixing board, respectively. The Control Display is headed by the title of the song, the version number, date, and time (all updated every time you save a piece of music). Below this are the metronome setting (variable from 0.5 to 480 beats per minute and adjustable in fine and coarse increments), a check box for enabling and disabling the MIDI Thru function, and a channelizer. If the channelizer is set, all data passed on the Thru line has its command bytes reset to play only on that channel. The next line is occupied by the beat counter (which can be step advanced by quarter and sixteenth notes with the mouse), transposing icons, and metronome. Music may be transposed 4 octaves up or down, by octaves or by half-steps. An interesting feature is the ability to exempt 1 track from the transposition. As always with MIDI recording, transposing a song and changing its tempo are independent functions, unlike tape. The metronome function can be turned off (it uses the

monitor speaker). It is tied very closely to the recording function, almost like a click track, so if a song has even one tempo change you will need an external metronome to provide the second tempo. The actual recorder controls are next. Starting at the left, they are Stop, Play, Pause, and Keep. The first three are fairly standard 'tape transport' controls. Everytime the program is in Play, any data on the MIDI In line will be stored in a holding buffer. Clicking on Keep stores the data on a selected track, as the buffer is written over on the next pass. The bottom line is a bar graph indicating total memory used.

The Track Display is basically a chart with the information for each track shown horizontally. At the extreme left is an arrow pointing to the current recording track. Each track is numbered and may be named. In the screen dump, these are not all tracks of one song, but different pieces of music condensed to a single track apiece. The first column after the track name contains an up or down arrow, depending on whether the track is turned on or off during playback. Other track information includes the end of data on a track (E), note activity, if a track has been changed (c), if it is protected (p), channel assignment, and percentage of memory. Tracks may be individually turned on and off, protected from accidental erasure, or rechannelled from their original MIDI channel to another, user selectably.

The rest of the program's functions are accessed from the pull-down menus. Selection of one of these options results in GEM dialog boxes. The File menu gives the normal disk options of Load, Save As, Update, Delete (which deletes a file from the current disk), Format Disk, Current Drive (to make use of a second drive or hard drive for storage), and Quit. Track functions are Name, Protect, Unprotect, and Delete, all self-explanatory. From the Edit menu it is possible to Name Song, Copy Track, Mix Track, Time-Correct Track, and Erase Song (only effects memory). Time-Correct allows correction of recorded timing data so timing errors are forced to the nearest note value selected; this in case the performance was a little sloppy. Notes can be quantitized to the nearest 32nd or 32nd triplet, or in grosser amounts if desired. For sending Mode or other similar MIDI commands to a synth, there is the MIDI menu. 'Mode' will pass commands to set channel selection, MIDI operating mode, and Local On/Off. 'Out' is for enabling and disabling the MIDI Clock data transmission. The last menu is Safety which allows you to toggle the warning alert boxes and the automatic file backup routine.

Overall, **EZ-Track** is easy to use, and has a fairly complete display of its activities. Even with its limitations, it is currently the strongest entry in the ST sequencer market. The documentation is clear, though anyone with experience with multi-track tape recording and the GEM interface won't need it very often. Having become hooked on using the mouse as a cursor controller for so much ST software, I am happy to see that almost every aspect of the program (except for alphanumeric input) can be driven by the mouse. I did have a minor problem in the beginning with the mouse rolling around during playback due to some rather strong bass vibrations; so much for mouse pads. 20 tracks is more than enough to do some complex layering of sounds and crossing parts. As a musician, I find it absolutely necessary that most of the music be entered real-time from an instrument keyboard, so this is a big plus. The only major complaint I have is the lack of editing capabilities. While it is possible to mix tracks, or delete them, this must be done to the entire track(s) involved. If during recording a mistake is made, there is no way to get rid of it other than to trash the whole track. It would have been very helpful if the effected track could be run up to the point of the error, and the rest of it deleted. This would save much time for non-virtuosoes. However, the way around this is to record everything in short takes at slow tempos. If a mistake is made, and a track tossed out, then not much effort is lost. Mix the tracks that are just right, set the tempo to the piece's normal speed, and voila!

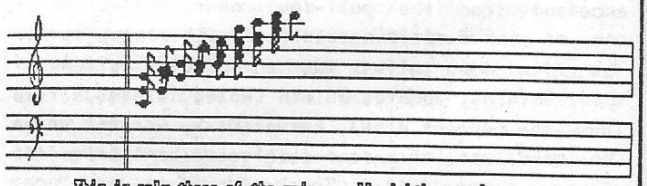
Hybrid Arts has established a policy for those that find the above approach to editing too painstaking and time-consuming. Soon to be released are **MIDITrack ST** ($375) and **MIDITrack Professional** ($575). The first will include 60 tracks, some real editing abilities like step-editing and auto-punch, and a hardware box providing sync-to-tape and external drum sync functions. The **Pro** version will have further edit options, plus the ability to read/write SMPTE time code and film click. One assumes that any files created with **EZ-Track** will be usable with the upper level products. Registered owners wishing to upgrade their software may do so for $15 plus the difference in retail price.

```
*********************************************
*                                           *
*                ST Music Box               *
*                                           *
* Type: Note editor                         *
* Level: Beginner                           *
* Requirements: MIDI synth for external audio *
* Features: step editing, music printout    *
* List price:                               *
* Publisher: XLENT Software                 *
*            P.O. Box 5228                  *
*            Springfield, VA 22150          *
*            (703) 644-8881                 *
*                                           *
*********************************************
```

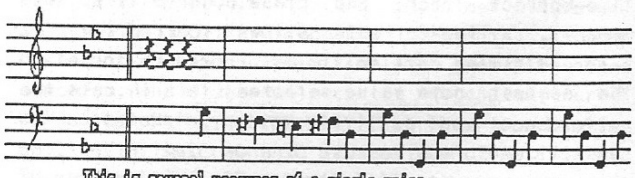
Picture, if you will, a modern day Bach or Beethoven. He no longer sits at the grand piano, one hand on the keys whilst the other messily scribbles notes with a quill; no, this brave, new composer's fingers dance across a computer keyboard, his face lit by the glow of the monitor as his melodies are channelled through his computer to a bank of MIDI synths. This would seem to be the vision XLENT Software had when they formulated the main concept for their program. **ST Music Box** is a step music notation editor; that is, it lacks the ability to record real-time from MIDI but can send MIDI data entered with either the mouse or ST keyboard. For those currently without a MIDI synth it does provide access to the internal ST voices. This feature will be somewhat glossed over in this review and more of the MIDI aspects discussed. The disk comes with several compostions you can load and play, and a program to print sheet music. **ST Music Box** runs in medium and high screen resolutions, though my disk was missing the monochrome version. As usual with XLENT, the program is not copy-protected. Also, as usual with data processing software, the data files generated with **Music Box** are not compatible with other authors' MIDI playback/sequencing/editing programs.

The program has two main screens which can be overlayed with menu windows, called by the function keys. The function keys are also used to make further selections in these windows. (There won't be any screen dumps accompanying this review as XLENT has seen fit to disable the standard ALT/HELP dump, and there is no way to exit the program, other than rebooting the computer, so none of the SNAPSHOT utilities will work.) The editor screen has a grand staff occupying its upper half, with mouse controls in the lower left and mouse keyboard for indicating notes in the lower right. There are eight voices for which you can enter music, but these are assigned in pairs to 4 seperate MIDI channels. A time signature can be picked (limited to 2/4, 3/4, 4/4, and 5/4) though it doesn't show on the staff; it only allows the editor to define measures. There is a full range of key signatures.

Music is entered in one voice, one note at a time. For each note a duration and an octave must be chosen. The notes appear on the staff but remain only until the measure is filled. There is a measure advance feature which automatically enters the notes in the memory buffer, but you can't enter more notes than allowed by the time signature even if the auto advance is toggled to off. You cannot call a window unless you are at the end of a measure. The number of symbols and beats already used in a measure, as indicated by the 'U' and 'B' in the icon box, is not always accurate. I couldn't find a way to display multiple voices or multiple measures which makes it very difficult to see what you're writing in context of what you've already written. There are a couple of nice features I'd like to see in all music editing software, though. You can insert patch change data by hand into every measure. You will have to experiment a little to match the instrument numbers in the program to the patch numbers on your synth. You can also change the tempo every measure. These are nice fine touches but don't overcome the limitations of the overall structure of the program.



This is only three of the voices, all eighth notes!



This is several measures of a single voice.

The player screen is basically composed of 8 'tracks' across which the notes play. You can finally see all 8 voices, notes or rests, at once, but they aren't in the context of a staff so the display of notes doesn't provide much more than

knowing a note is in a specific voice. At the far right of the screen are a measure count and indicators of what patch number is currently assigned to which channel. The top of the screen shows portamento control status (which is implemented in the program rather than effecting the portamento or glide function on your synth), channel status, and tempo. One strange thing about tempo values (on both screens) is that smaller values are faster tempos, and larger values, slower; this is backwards from most music notation where higher numbers indicate speedier tempos.

The print program essentially does a dump of the top half of its screen, giving a slightly larger than normal notation. There is a text editor which gives you access to the ST font options, thickness, skewing, etc. You may add comments or lyrics in all shapes and sizes. You can also add the top half of any 32K picture file of appropropriate resolution, and save files in a format usable with XLENT's **Rubber Stamp ST** or Typesetter ST, though I don't see much room on sheet music for extraneous graphic material. You can display notes by the screenful and print each screen or dump entire pieces. One of the limits here seems to be that you can't print a whole line of multiple voices; they are printed one measure per line. Even then you must make certain that some voices don't block out other voices; rests are overlayed on notes, and several stacked eighth notes may wind up looking like sixteenth or thirty-second notes. Single voices always print the whole grand staff even if they only make use of one of the staves. The print program might be more valuable as an option integrated into the **Music Box**, rather than as a seperate program.

XLENT's software seems to fall into that category where users either love it or hate it. I think, in this case, their approach is interesting, but the software is hampered in places by a lack of understanding of what musicians want. Using the program is further slowed by its dragged-out screen redraw and its insistence on emptying, and reacting to, the ST's keyboard buffer. Some things are merely annoying, like having to call the disk menu once to get a directory, then again to load a composition from the disk. Others I find downright unworkable such as not being able to display more than one measure at a time. Certain 'programmer's compromises' are questionable; for example, having a dotted quarter note play through MIDI as a quarter note and an eighth note, rather than as if the note was held, does not strike me as useful. Not being able to enter music from an instrument makes this at best a composing tool, which possibly can be understood if viewed within the framework of XLENT's graphic editors. I think **Music Box** in the long run, with some further refinement to its abilities, could prove to be very helpful in producing simple scores and other types of sheet music, but this will require, of course, they provide the routines to convert files between the varying MIDI software companies' data storage formats.

More MIDI Notes from page 13

has been waiting, and a complete(?) list of MIDI software and vaporware with their companies and addresses. Remember, it is up to you to let them know what it is you want. Until then, merry MIDI.

Late flash! After several months of trying to make long distance connections with the people at Key Clique but never succeeding, it was a pleasant surprise to get a demo copy of **SYS/EX** in the mail yesterday. This is a generic patch/sequence librarian that you can expand to cover all instruments or other devices that comply with the MIDI system exclusive protocol. Though it won't help you edit your patches, **SYS/EX** won't become obselete as you upgrade your equipment. To be reviewed in December or January.

COPY II ST™

BACKUP PROTECTED SOFTWARE FAST.

From the team who brought you COPY II PLUS (Apple), COPY II PC (IBM) and COPY II MAC (Macintosh) comes a revolutionary new copy program for the Atari 520 and 1040 ST computers.

- Copies many protected programs—automatically. (We update COPY II ST regularly to handle new protections; you as a registered owner may update at any time for $15 plus $3 s/h.)
- Supports single and double sided drives.
- Includes both a fast sector-based copier and a true bit copy mode for protected disks.

Requires an Atari 520 or 1040 ST computer with one or two drives.

Call 503/244-5782, M—F, 8—5 (West Coast time) with your VISA in hand. Or send a check for $39.95 U.S. plus $3 s/h, $8 overseas.

$39.95

Central Point Software, Inc.
9700 S.W. Capitol Hwy. #100
Portland, OR 97219

Central Point Software INCORPORATED

Backup utilities also available for the IBM, Apple II, Macintosh and Commodore 64.
This product is provided for the purpose of enabling you to make archival copies only.

FORTRAN

as a Second Language

Pro-Fortran's GEM Interface

by Mark Lovell

I don't know how many of you are using the Prospero FORTRAN compiler, but if you are, you have probably noticed a rather large problem: the GEM interface. Prospero provides a full GEM library and documentation with their compiler, but the library is full of bugs. If you have read my August column, you know that I have had to work around some of the limitations in it. No real major problems showed up: I could not rotate text, or put a "close window" icon on the plotting window. I could live with these limitations, but I found others recently that were unacceptable, and more were popping up every day.

Prospero made some assumptions when they packaged their GEM library. They thought that they had provided every function and subroutine a user would need. Of course, this assumed that all the routines worked.

My first idea was to fix the buggy routines. I looked real hard at the Prospero documentation, and found several common blocks for passing arguments to GEM and one subroutine called "AESIF," and no clues about how to get to the GEM interface directly. It seemed like I could not fix the problems myself, but I did find a solution.

As with most compilers, the option of rewriting a library is always available, but in doing this I found some new problems. The most major was learning how to use Prospero's assembly language interface. A second was looking at GEM from the assembly level.

If you plan to do anything like this yourself, I would suggest that you get hold of several items first. The first item is a compatible assembler. Prospero FORTRAN uses the GST linker, which uses a binary file that is quite different from those used in the Atari developer's package. Luckily the GST linker has become popular; it is used by all of Metacomco's products (Lattice C being the most popular), and all of Prospero's products as well. Metacomco makes a macro assembler that is perfect for this purpose.

A second item that you might find useful is a 68000 machine code reference manual. The documentation that comes with a macro assembler rarely covers the machine itself. Even if you feel comfortable with the 68000, a good reference for the machine would not hurt. I recommend "68000 Assembly Language Programming," put out by Osborne/McGraw-Hill. It is not only a good reference, but a good tutorial as well.

The last item that you will need is a good GEM reference. At the moment, there are only two sources: the Atari developer's package, or Abacus's "GEM Programmer's Reference." I can't say that I highly recommend the Abacus book -- it has too many misprints, and omits many routines -- but at $19.95, it is a lot cheaper than Atari's solution. I have used the Abacus book for this project.

FORTRAN cannot directly call GEM. This is because GEM requires that certain registers be loaded with information and that a machine language "TRAP" instruction be executed. I have written two assembly language programs to remedy this problem. Only two assembler routines were necessary because GEM has only two methods of being called, one for the VDI routines and one for the AES routines. I put a minimal amount of code in assembly language, so the final result would be as understandable as possible. The way I have set things up, a FORTRAN routine can load the necessary parameters into arrays, and pass the arrays to the assembler routine to pass on to GEM. This way most of the

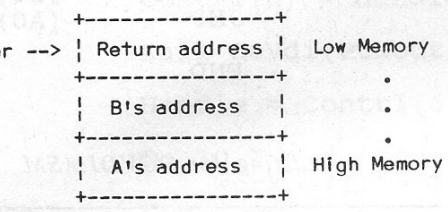
new GEM library can be in FORTRAN, which is quite a bit more understandable than assembler.

Interfacing FORTRAN to assembly language:

When Prospero FORTRAN calls a subroutine, it puts the addresses of the arguments on the stack. So if we have a subroutine call like this:

```
Call Add(A, B)
```

the stack will look something like this:



To access either argument from assembly language, you can use its address. When the assembler routine returns, there is a little cleaning up to do. It is the responsibility of the called routine to remove the arguments from the stack before the routine returns. Prospero recommends the following sequence:

```
MOVE.L    (SP)+,A0    Pop return address off stack
ADDA.W    #8,SP       Remove adr of arg from stack
JMP       (A0)        Goto return address
```

Since there are two arguments in the above example, "8" (2 times the size of an address) is added to the stack to remove the arguments. The "8" would be replaced in other cases the number of arguments times 4, to be correct. With this information we can now write an assembly language routine to do just about anything.

Interfacing assembly language to GEM:

Interfacing with GEM is not as easy in assembler as it is in a high level language. With VDI routines there are five arrays used for communication:

Name	Use
Control	Passes the # of the cmd, and cmd info
Intin	Numbers required for the command
Ptsin	X, Y points required for the command
Intout	Number output by the command
Ptsout	X, Y points output by the command

```
      Common      /Contrl/ Contrl(0:11)
      Integer*2   Contrl

      Common      /AGloba/ AGloba(0:14)
      Integer*2   AGloba

      Common      /Intin/ Intin(0:127)
      Integer*2   Intin

      Common      /Intout/ Intout(0:127)
      Integer*2   Intout

      Common      /AddrIn/ AddrIn(0:63)
      Integer*4   AddrIn

      Common      /AddrOu/ AddrOu(0:63)
      Integer*4   AddrOu

      Common      /Ptsin/ Ptsin(0:127)
      Integer*2   Ptsin

      Common      /Ptsout/ Ptsout(0:127)
      Integer*2   Ptsout
```

Listing for GEMCOMM.FOR

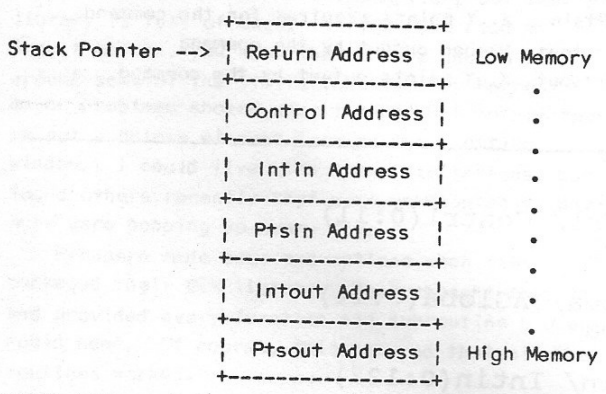
Each of these arrays is required for every VDI command. Rather than defining them in each routine, I have set up a common area for each of them. The common area definition is shown in the file "GEMCOMM.FOR."

When calling GEM from assembler, the address of a parameter block must be passed in register D1 and the library number ($73) must be passed in D0. After the registers are set up, a "TRAP 2" instruction must be executed.

The format of the parameter block is rather simple -- it consists of the addresses of the five parameter arrays. If we pass the arrays as arguments to the assembler routine, their addresses will be available for passing along to GEM. Since FORTRAN pushes the address of the first argument on the stack first, the first argument will be the last in the parameter block. If we call the assembler routine like this:

```
Call DoVdi(Ptsout, Intout, Ptsin, Intin, Control)
```

Then the addresses of the arguments will end up on the stack like this:



which happens to be exactly the way GEM wants the parameter block to be set up. The address of the paramenter block is now the value of the stack pointer + 4, since addresses are 4 bytes long. My implementation of "DoVdi" is listed in "DOVDI.ASM."

The AES interface works similarly. It requires six parameters:

Name	Use
Control	Same as for VDI
Global	Stores GEM constants
Intin	Same as for VDI
Intout	Same as for VDI
Addrin	Addresses sent to AES
Addrou	Addresses received from AES

The library number for the AES is "$C8." The AES assembler routine is called this way:

```
        XDEF    DOVDI
VDI     EQU     $73
        TEXT
DOVDI   MOVE.L  SP,D1
        ADDQ.L  #4,D1
        MOVEQ.L #VDI,D0
        TRAP    #2
        MOVE.L  (SP)+,A0
        ADD.L   #20,SP
        JMP     (A0)
        END
```

Listing for DOVDI.ASM

```
        XDEF    DOAES
AES     EQU     $C8
        TEXT
DOAES   MOVE.L  SP,D1
        ADDQ.L  #4,D1
        MOVE.L  #AES,D0
        TRAP    #2
        MOVE.L  (SP)+,A0
        ADD.L   #24,SP
        JMP     (A0)
        END
```

Listing for DOAES.ASM

```
Call DoAes(Addrou, Addrin, Intout, Intin, Global,
Control)
```

The routine "DoAes" is listed in the file "DOAES.FOR."

Using the new interface:

Each array must be loaded with the correct values, for the GEM call to be made. After the call is made, any returned values must be moved from the arrays to the proper variables. The information needed to make the various GEM calls is provided in the Abacus GEM reference manual. The two assembler routines could be used directly in a program, but for the sake of readability, I would

```
      Subroutine Vopnvw(Workin, Handle, WorkOu)

      Include           'C:GemComm'

      Integer*2         Workin(0:10)
      Integer*4         Handle
      Integer*2         WorkOu(0:56)

      Integer*4         I

      Contrl(0) = 100
      Contrl(1) = 0
      Contrl(3) = 11
      Contrl(5) = 0
      Contrl(6) = Handle

      Call DoVdi(Ptsout, Intout, Ptsin, Workin, Contrl)

      Handle = Contrl(6)

      Do 10 I = 0, 44
          Workou(I) = IntOut(I)
10    Continue

      Do 20 I = 45, 56
          WorkOu(I) = ptsout(I-45)
20    Continue

      End
```

Listing for VOPNVW.FOR

```
      Subroutine VPline(Handle, Count, XY)

      Include           'C:GemComm'

      Integer*4         Handle
      Integer*4         Count
      Integer*2         XY(0:*)

      Contrl(0) = 6
      Contrl(1) = Count
      Contrl(3) = 0
      Contrl(6) = Handle

      Call DoVdi(Ptsout, Intout, XY, IntIn, Contrl)

      End
```

Listing for VPLINE.FOR

not recommend it. I think the best way to use them is to recreate the GEM library provided with Prospero's FORTRAN. The format of their library is simple and easy to follow. The routines for "VOPNVW" and "VPLINE" are listed as examples.

As a bonus for disk subscribers, I have included the source for a complete GEM library with a calling scheme identical to the format documented in the Prospero manual.

Late Breaking News:

I have received a letter from Prospero in England. They insist that their compiler and its libraries have been thoroughly debugged. News from another British company indicates that there are some diffences in the British ST's (both keyboard and monitor are different). This may be the source of the problems I have had. Prospero seemed quite interested in finding out about possible errors in their system; so, if you find any new ones, send them to me, and I will collect them and send them on to Prospero.

Diskette Control

by David Bessey

Overview

This is the first part of a series of articles discussing diskette drives, diskette formats, software, and drive connections. The approximate contents of these articles, which will appear intermittantly, are as follows:

1. "Density", and formats
2. Programming of the 1772
3. Drive connections and signals
4. Connecting a 5.25" drive to the ST
5. Writing your own I/O drivers

When this series is done you should be able to read and write to micro-(3.5") and mini-(5.25") diskette drives using any sector size you need. I don't use copy-protected diskettes, so I can't discuss them.

Density

"Density" is a term from physics which means "mass per unit volume"; Density is measured in grams-per-cubic-centimeter. One CC of iron weighs more than one CC of water because it is more dense than water. "Bit packing density" refers to how much information can be recorded on (or in) some recording medium. This is bits-per-square-centimeter for a magnetic-surface device such as tape, disk, or diskette. Because recording is done in a linear fashion (you can't record an area), the "recording density" is measured as bits-per-cm along the tracks and as tracks-per-centimeter going across the tracks (see Fig1). This reduces the overall density because of the space "wasted" for guardbands between the tracks.

Everyone has heard of "single density" and "double density" by now, but "What is it?" you ask, and "What about 'quad density'?". This is a long story...

A diskette drive, like an audio tape recorder, has a "frequency respone" range. This means that the reproduced signal has, over this frequency range, a relatively constant amplitude. The span from 20Hz to 20kHz is approximately ten octaves, a range which is extremely difficult to achieve. The signal processing needed to obtain this frequency range and the magneting recording process itself both cause excessive amounts of distortion to the recorded signal. The flat "frequency response" graphs shown on a machine's data sheet shows what has been bought by creating this distortion--this graph cannot show the distortion of the waveshape of the signal.

Digital signals, such as are found in disk and diskette drives, MUST have their waveshapes accurately reproduced. In addition, each bit must be distinct from its neighbors and it must occur at the right time; this is common info to computer people. What most people don't know is that the distortion referred to above distorts all three of these necessary factors. To minimize the distortions -that is, to minimize BER(Bit Error Rate)- three steps are taken:

1. NO signal processing.
2. channel coding to minimize bandwidth.

3a. BCC(Block Check Code) to detect errors. (or sometimes)

3b. ECC(Error Correction Coding) note: These codings are not cryptography.

The third item requires its own article, and the first requires no discussion. Bandwidth coding is what most affects recording density. There are many coding schemes for magnetic recording, but only three will be discussed: NRZ (Non Return to Zero), Manchester, and Miller.

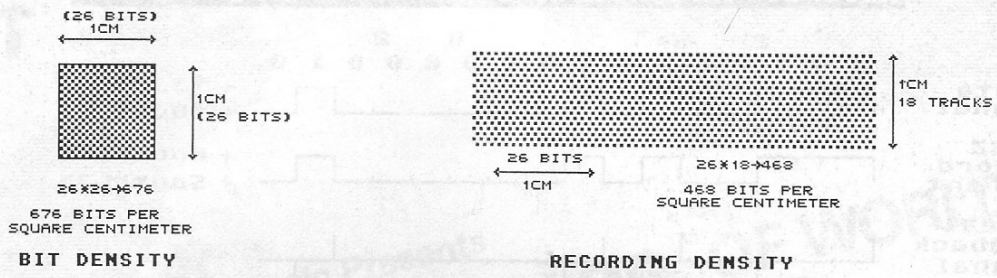


Figure 1. Recording Density is measured as bits-per square-centimeter along the tracks and tracks-per-centimeter going across the tracks.

"In 'NRZ(level)', zeroes and ones are recorded as positive and negative levels so that transitions only occur whenever the data bits change. ... For long strings of zeroes [and ones] ... the maximum DC content is unity".(1) When the recorded signal is DC, the playback head produces nothing because it is sensitive only to a CHANGE in recorded signal. When these "dead spots" occur, the playback circuits lose their clock-synchronization and data are lost. NRZ IS usable for low data-rate applications; it is used because it requires only very simple circuitry.

"Manchester coding and its variants are known by many names: bi-phase, double-frequency, phase-encoding, frequency-shift-coding and, quite inaccurately, frequency modulation."(1) Manchester is used for "single density" diskette data recording. It has zero DC content, and is self-clocking. Its drawback is that it requires two bit-periods to record each data bit. It IS easy to use and easy to understand. (see fig2.)

"Miller code is known by two other names: delay-modulation and (even more confusingly) modified-frequency-modulation (MFM)"(1) The Miller coding method is used for "double density" diskette recording. Its advantage relative to Manchester is that it records a data bit for every bit period. This is why it is mis-called "double density". Its disadvantages are its complexity and its DC content of approximately thirty-three percent. Its lower signal-to-noise ratio and its complexity make it difficult to use, but the payoff is increased data storage.

It should be obvious that "quad-density" doesn't exist. If it did, it would be capable of recording TWO data bits during EACH bit period. There ARE (for radio) methods of transmitting two or four bits per bit period, but they are analog methods and won't work for disk/diskette/magtape data recording.

Digital data recorders use saturation recording. This means that the magnetic force recorded on the medium is as strong as it can be. Amplitude variations are eliminated during the playback process to improve data reliability; this is why the analog data compression methods don't work for digital recording systems.

The last pattern shown in Fig2 shows how the Miller code stores twice as much data as the Manchester code in a given amount of space. When the smallest bit-cell period of the Miller code is made equal to the smallest bit-cell period of the Manchester code, it can be seen that the Miller encoded data require half as much space as the Manchester encoded data. If the same space is used, the Miller code will store twice as much data as the Manchester code. The Miller-squared code, which I didn't discuss, also records one data bit in each bit period. The Miller-squared code eliminates the DC content of the basic Miller code at the expense of more-complex circuitry.

The limit on what can be recorded is due to (a)the record process, (b)the recording medium, and (c)the playback process.(2) Fig2 shows there is a vast difference between the ideal and the actual playback respones. The playback is affected by ALL the recording distortions even though the playback process itself is only a partial contributor to the total distortion.

As can be seen, if the data bits are too close together they will tend to merge and form patterns similar to the troublesome NRZ patterns. Some errors in the magnetic recording process actually cause some of the data bits to move to positions earlier or later than where they should be. This timing distortion can cause bits to merge: if one bit is late and the next early, they can be mis-detected by the playback circuitry, and... (Ta da...) errors result. This effect can be compensated for somewhat by a write precompensation system in the recording circuitry and by pulse-narrowing circuits in the playback amplifiers(3).

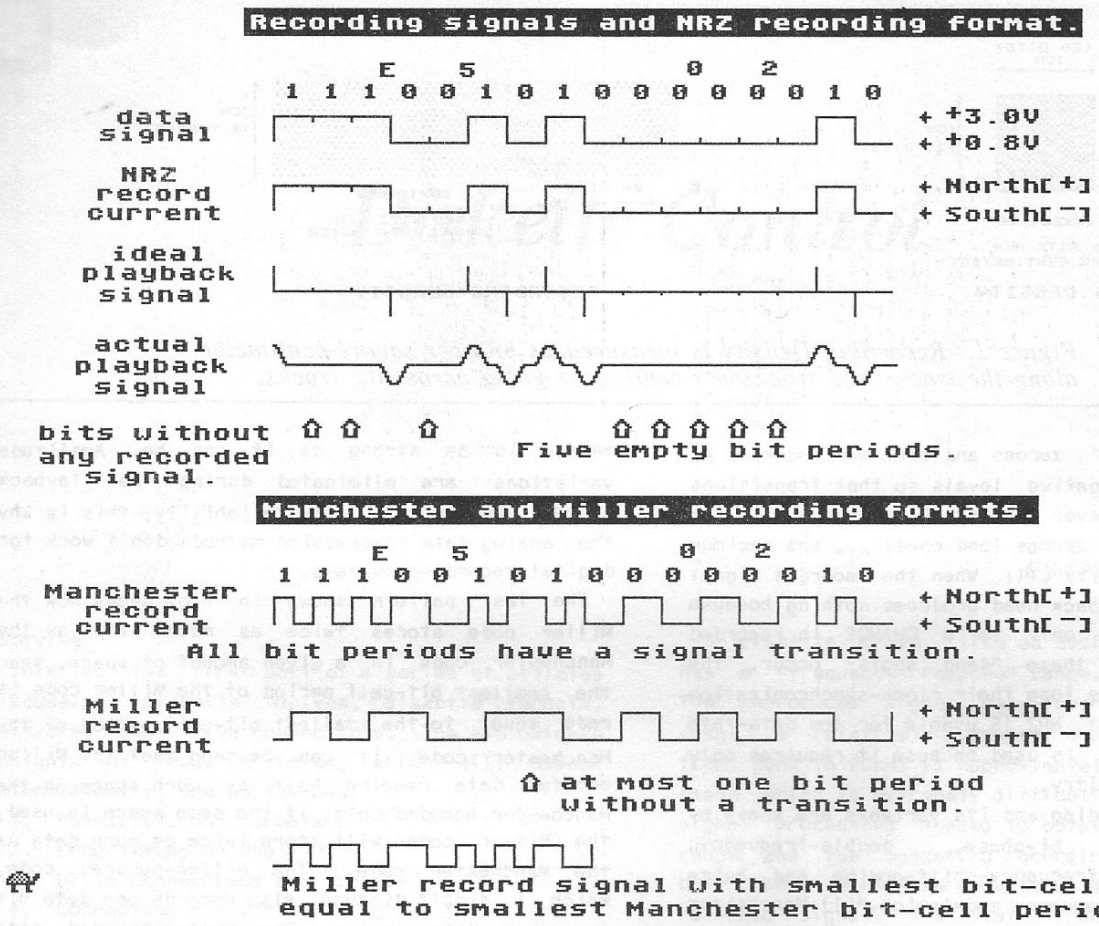


Figure 2. Manchester Coding is used for single density recording. Miller Coding (modified frequency modulation) is used for double density.

Formats

Every once-in-a-while someone asks me "Is the IBM [diskette] format 128B or is it 256B?". The answer is "Yes.", of course, because the IBM diskette data format specification allows the use of 128, 256, 512, or 1024 bytes per sector. Some of the overhead information recorded for each diskette sector reveals the sector's length. At this time I will describe the diskette formats from the diskette's point of view and sometime in the future I will discuss how to create your own diskette formats.

ST diskette format data for one track...

#B	Val	Description
60	4E	Index pulse trailer
§§12	00	IDAM clock-synch
03	A1	Addr Mark lock-in marker
01	FE	ID Address Mark
01	xx	(00:4F) Track
01	xx	(00:01) Side
01	xx	(01:09) Sector
01	02	Sector size (512B)
02	XX	CRC bytes (Cyclic Redundancy Check)
22	4E	IDAM trailer
12	00	DAM clock-synch
03	A1	Addr Mark lock-in marker
01	FB	Data Address Mark
512	xx	The DATA
02	XX	CRC bytes
40	4E	Data sector trailertt
1401	4E	Track trailer

The data inside the braces is repeated for each sector. The data before and after occur only once per track. This will be explained in detail when track formatting is discussed. Items marked "xx" are user-supplied while the items marked "XX" are supplied by the 1772. The two CRC bytes are a sixteen bit code used to detect errors in the data on playback. It cannot correct any errors and can't even detect all errors.

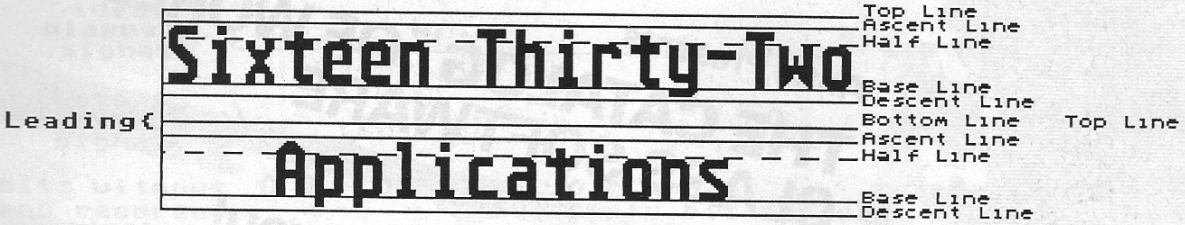
References

(1) "Optimal codes for digital magnetic recording" by J. C. Mallinson and J. W. Miller; The Radio and Electronic Engineer, Vol 47 No. 4, pp. 172:176, April 1977. The Miller who co-authored this paper

Continued on page 34 -->


ST
Antic Presents
THE CATALOG™ OF WORLD CLASS SOFTWARE
Delivered directly to you!
CALL NOW—IT'S FREE!
800-443-0100 ext. 133
24 hours-a-day for credit card holders. (Continental USA And Hawaii)
START—The ST Quarterly. Get your premier issue of START—a power users quarterly with disk. Single copy $14.95. Yearly $59.95
SOFTWARE THAT FINISHED FIRST AROUND THE WORLD
Tom Hudson's CAD 3-D (ST0214) $49.95
A-CALC by Kuma (ST0212) $59.95
Flash (ST0220) $39.95
WORLD CLASS SOFTWARE
Expert Opinion (ST0219) $99.95
Maps and Legends—The Cartographer (ST0202) $34.95
A-Seka by Kuma (ST0216) $34.95
A-RAM by Kuma (ST0215) $19.95
GST-C (ST0217) $79.95
GST-ASM (ST0218) $59.95
MCC Pascal by Metacomco (ST0206) $99.95
MCC Macro Assembler by Metacomco (ST0205) $79.95
Lattice C (ST0207) $149.95
C.O.L.R. Object Editor (ST0201) $29.95
Disk Doctor (ST0211) $29.95
Red Alert (ST0223) $24.95
Star Struck—The Astrologer (ST0222) $24.95
Mom and Me (ST0204) $19.95
Murray and Me (ST0203) $19.95
Ultima I by Sierra (TH9001) $49.95
Ultima II by Sierra (TH9002) $49.95
WE WANT TO HEAR FROM YOU TODAY!
Enclose your check or money order (no cash) with the items you are purchasing clearly listed on a sheet of paper. Please print the title as it appears complete with the (STO #) and price. ADD $3.00 SHIPPING, California residents please include 6.5% sales tax. INCLUDE YOUR NAME, ADDRESS, ZIP CODE, PHONE NUMBER.
Send your order to:
THE CATALOG
524 Second St.
San Francisco, CA 94107
Products from "THE CATALOG" are backed by the "Antic Customer Satisfaction Guarantee".
WE ALSO SELL XL/XE SOFTWARE. For more information on all of "THE CATALOG" products write to the address above attention "Customer Service" OR CALL TOLL FREE 800 443-0100 ext. 133 for a FREE 40 page color catalog.
•Retailers—Inquire about our Dealer prices. Call (415) 957-0886.

A Step FORTH



Example of VST_Alignment

The Lesser of Two Evils- VDI Part 2

by Albert Lew

After much deliberation, I have decided to postpone a discussion of the AES until another month. The less desirable features of VDI, such as inputs, inquiries, and text, are probably the lesser of two evils, as they are easier to understand and use. Admittedly, although the mouse inputs used by VDI are primitive, and the inquiry functions are quite dry, text on the ST is a rich, interesting topic. But alas, I am getting a bit ahead of myself. The two general types of functions described in last month's column were output functions, which make things happen on the screen, and attribute functions, which set the parameters for the output functions. Text, of course, is an output function. The two important ST input functions are those used for getting the position of the mouse and inputting strings. String input does not work that well, but obtaining mouse coordinates is quite elegant. Inquiries are basically the opposite of parameters. By using inquiry calls, the programmer can determine the state of various VDI attributes, thus the name inquiry. With knowledge of these three techniques, VDI programming becomes even more enjoyable.

This column will provide examples for 4x FORTH, Forthmacs, and Multi-FORTH. VDI works strangely under GEM, and in Mach 2, it is impossible to type in input examples from within the Forth. String handling in Mach 2 is also much more primitive; an explanation of creating packed strings would be in order, and that is an entirely different subject. This difficulty with strings makes text output just slightly difficult. However, the inquiry features mentioned can be used with Mach 2; to use them type 'call' and then the name of the 4xFORTH Forthmacs name minus the '_' (underline). For those who did not read the September '86 A Step Forth, this column is simply not going to make sense. Read it first -- after all, this is VDI Part 2 and none of the examples will work unless you type in the initialization procedures described in that article.

Text in all its Glory

There are two kinds of text supported by VDI: graphics text and justified graphics text. Justified graphics text output is by far the more typeset-looking, whereas plain graphics text is dully monospaced. However, both text types use the same attributes and inquiries. The graphics text call is v_gtext (VTextDraw), and the justified text call is v_justified. (Multi-FORTH equivalents to the Forthmacs/4x FORTH/DRI standard are in parentheses.) The parameters for graphics text are the x and y starting coordinates and the string to be printed. In addition to the parameters graphics text needs, justified graphics text requires the intended length in pixels of the text, and specifications as to whether GEM may modify character and or word spacing. Both commands, of course, require a handle when used from within Forthmacs; see last month's column for an explanation of VDI handles.

The text attributes are VST_HEIGHT (VTextHeight), VST_POINT (VTextSize), VST_ROTATION (VTextDirection), VST_FONT (VTextFont), VST_COLOR (VTextColor),

VST_EFFECTS (VTextStyle), and VST_ALIGNMENT (VTextAlign). VST_HEIGHT sets the height in pixels for all future text called from either the graphics or justified text commands. Also required in the VST_HEIGHT call (only for Forthmacs) are the address of the variables charwidth, charheight, cellwidth, and cellheight. These variables simply have to be created by the programmer (i.e. type: variable charwidth variable charheight, etc.) and when the routine is called, the values of the character width, etc. end up in the variables. Character width and height are the dimensions of the printed characters in pixels; and the cell dimensions are for the space that the character, including surrounding space, takes up. By returning the values actually set, VDI tells the programmer to what degree his or her instructions have been carried out. In Multi-FORTH and 4xFORTH, this feature is not accessible; the only parameter that needs to be passed is the text size. The attribute VST_POINT takes a point size instead of a pixel height, with a point being approximately 1/72". This doesn't mean anything, of course, because an 8 point selection on a monochrome monitor is much smaller than a similar 8 point selection in low res on a color monitor. The same variables used with VST_HEIGHT are used with VST_POINT. VST_ROTATION offers the magical ability to turn the text at any degree from 0 to 360. Unfortunately, doing this is still magical, because text will only tilt at four angles: 0, 90, 180, and 270 deg. The angles are specified like those for arcs and elliptical arcs. Thus, to rotate text 90 degrees, 900 would have to be the parameter (along with the handle in Forthmacs) passed to VST_ROTATION. VST_FONT, unfortunately, does even less than VST_ROTATION, which is to say, absolutely nothing. There is only one font available on the ST, so VST_FONT is of no use. VST_COLOR selects the color index for the text to be printed in. The color index selection is just like that for lines and markers and fills. You can choose from black and white in monochrome, one of four colors in medium res, and one of sixteen colors in low res. VST_EFFECTS has 63 possible combinations, and is controlled by sending an effect number as a parameter to VST_EFFECTS. Setting bits 0-5 will result in changing thickness, intensity, skewing, underlining, outlining, and shadowing respectively. Thus, if the effect number is 63, the text printed will be thickned, intensified, skewed, underlined, outlined, and shadowed (1+2+4+8+16+32). Some people might even call it illegible... Finally, there is the attribute VST_ALIGNMENT, whose first parameter determines justification/horizontal alignment (0 = left, 1 = center, 2 = right), and whose second paramter determines vertical alignment (0=baseline, 1=half line, 2 = ascent line, 3 = bottom, 4 = descent, 5 = top). Basically, the vertical adjustment, in order of increasing height, is bottom line, descent line, base line, half line, ascent line, and top line. If all these parameters and commands seem confusing, the following tables and examples should clear things up.

Table 1: TEXT OUTPUT

Op. Code	Name	Syntax
8	V_GTEXT (VTextDraw)	(x y string --)
11, id=10	V_JUSTIFIED (N/A)	(x y string length word-space char-space --)

x and y: coordinates of the lower left hand part of the text to be printed.
string: address, then length of a packed string is placed on the stack. (address only for 4xFORTH)
wordspace: interword spacing. 0=off. Non zero=on. When word spacing is "on," GEM can (and usually will!) alter spacing between words.
charspace: intercharacter spacing. 0=off. Non zero=on. When character spacing is "on," GEM can (and usually will!) alter spacing between individual characters.
Remember: always use a handle (just the number '1') as the first parameter in a Forthmacs VDI call.

Table 2: TEXT ATTRIBUTES

Op. Code	Name	Syntax
12	VST_HEIGHT (VTextHeight)	(height charwidth charheight cell-width cellheight --)
107	VST_POINT (VTextSize)	(point charwidth charheight cell-width cellheight --)
13	VST_ROTATION (VTextDirection	(angle --)
21	VST_FONT (VTextFont)	(font --)
22	VST_COLOR (VTextColor)	(index --)
106	VST_EFFECTS (VTextStyle)	(effect --)
39	VST_ALIGNMENT (VTextAlign)	(horin vertin horout vertout --)

height: height in pixels of future text.
charwidth: a user intialized variable that returns the character width selected by GEM. Not used by Multi-FORTH or 4xFORTH.
charheight: a variable that returns the character height selected by GEM. Not used by Multi-FORTH or

4xFORTH.
cellwidth: a user initialized variable that returns the cell width selected by GEM. Not used by Multi-FORTH or 4xFORTH.
cellheight: a user initialized variable that returns the cell height selected by GEM. Not used by Multi-FORTH or 4xFORTH.
angle: angel of rotation -- 0=0, 900=90, 1800=180 deg, etc. Only acceptable values are 0, 900, 1800, and 2700.
font: # of font to be selected. Does not work because there is only one font!
index: color index to be selected. 6 would select the sixth color from sixteen choices in low res. 0=white and 1=black in monochrome.
effect: text effect. Setting a bit to 1 selects effect.

```
Bit 0 = Thickening
Bit 1 = Intensity
Bit 2 = Skewing
Bit 3 = Underlining
Bit 4 = Outlining
Bit 5 = Shadowing
```

horin: horizontal alignment/justification desired to be selected.

```
0 = left justified (default)
1 = center justified
2 = right jusitified
```

vertin: vertical alignment desired to be selected.

```
5 = top line                TOP
2 = ascent line              ^
1 = half line               /|\
0 = base line (default)      |
4 = descent line             |
3 = bottom line           BOTTOM
```

horout: a user initialized variable that returns the horizontal alignment/justification value selected by GEM. Not used by Multi-FORTH or 4xFORTH.
vertout: a user initialized variable that returns the vertical alignment value selected by GEM. Not used by Multi-FORTH or 4xFORTH.
Remember: MultiForth words are in parentheses.
Notes: VST_HEIGHT may appear as vsr_height in Forthmacs. Change it to vst_height. Also, vst_effects in Forthmacs may call the incorrect opcode. If so, change the code "21 set-one" to "106 set-one" in ForthEmacs.

EXAMPLE 1 (Forthmacs)

Type:

```
string-array stapps ," ST Applications"
end-string-array
variable charwidth
variable charheight
variable cellwidth
variable cellheight
1 40 charwidth charheight cellwidth cellheight
vst_height
charwidth ? charheight ? cellwidth ? cellheight ?
1 4 vst_effects
1 1 vst_color
erase-screen
1 320 200 xy.scale 0 stapps v_gtext
erase-screen
: expandit!
    240 0 do
          1 320 i 2/ - 200 xy.scale 0 stapps i 0 1
          v_justified
          3 +loop
;
1 2 vst_color
erase-screen expandit!
```

-->Hints: If Forthmacs does not know what xy.scale is, the code for xy.scale is in the September A Step Forth. Also, if Forthmacs does not know what vst_height is, try using vsr_height instead.

EXAMPLE 2 (4x FORTH)

Type:

```
TOS$ STAPPS ST APPLICATIONS"
40 VST_HEIGHT
4 VST_EFFECTS
1 VST_COLOR
CLEAR
320 200 XY.SCALE STAPPS V_GTEXT
CLEAR
: EXPANDIT!
    240 0 DO
          320 I 2/ - 200 XY.SCALE STAPPS I 0 1
          V_JUSTIFIED
          3 +LOOP
;
2 VST_COLOR
CLEAR EXPANDIT!
```

-->Notes: You will not be able to read the text; 4x FORTH does not blank out each line after writing it

EXAMPLE 3 (Multi-FORTH)

Type:

```
CREATE STAPPS ," ST Applications"
40 VTextHeight
4 VTextStyle
1 VTextColor
ClearScreen
320 200 XY.SCALE COUNT STAPPS VTextDraw
```

-->Hints: The justified "special effects" text cannot be done on Multi-FORTH. Also, type in the code for Forthmac's XY.SCALE from the September A Step Forth.

What you see in examples one through three are the italicized words "ST Applications." The variables in

the Forthmacs are displayed so that you get an idea of how close GEM will come to doing what is expected. Part of the code does not work in Multi-FORTH because Multi-FORTH does not support justified graphics text. This part of the demo performs a word expansion in the middle of the screen with the text "ST Applications." The compromise between 4xFORTH/Multi-FORTH code and Forthmacs VDI convention should be clear. Although Multi-FORTH and 4xFORTH are easier to use and require less code, the user also has less control over what happens and thus obtains a poorer understanding of VDI.

Two Interesting VDI Input Functions

VDI input functions allow input from the user to the system. Many of the inputs are not direct real-world inputs, but two that are useful are the locator (mouse) and string input functions. These inputs, like the other two inputs, (which I will not discuss) valuator and choice, have two states. The most useful state is the request state. When the computer is requesting, it waits until the input of the device desired is entered before proceeding. In the sample state, VDI will simply look at whatever is going on at the given moment, whether the user is ready or not. To set the device type (locator, valuator, choice, or string) and the input mode (request or sample), VDI provides the programmer with the command VSIN_MODE (input functions are handled on a much higher level in Multi-FORTH, which has no need for VSIN_MODE). The first parameter VSIN_MODE takes is for the input device, and the second for the input mode. The correct device and mode must be set before invoking any of the eight basic (four times two) input functions. If the mode is set incorrectly, havoc will result when the input routine is called...

To request a locator input, VSIN_MODE must be set to device one, mode one. After that is taken care of, the programmer passes the starting coordinates of the mouse cursor and the addresses of the user-initialized variables xout, yout, and term to VRQ_LOCATOR. VRQ_LOCATOR will terminate ONLY WHEN either a key or mouse button has been pressed. The variable term will hold the type of termination; 30 if VRQ_LOCATOR was terminated by the left button, 31 if VRQ_LOCATOR was terminated by the right button, and a corresponding ascii value if VRQ_LOCATOR was terminated by a keypress. Xout and yout, of course, will hold the x and y coordinates of the mouse cursor at the moment of termination.

In order to start a string request input, VSIN_MODE parameters must be set for device four, mode one. The parameters for the the string request input, or VRQ_STRING, are the maximum length of the string in characters, the echo mode, the array containing the coordinates of the start of the echo, and the address of the string which will store the input. The echo mode does not really matter, since characters do not normally echo, and therefore, the importance of the echo array is nil as well. Unfortunately, the string input is a C string (different from a Forth string), which makes it difficult to manipulate. The two events that will stop VRQ_STRING are a carriage return or a full buffer, which is when the user has typed in more characters than were specified in the maximum length parameter. Of course, a crash would stop things for good just as easily...

Table 3: INPUT FUNCTIONS

Op. Code	Name	Syntax
33	VSIN_MODE (N/A)	(devicetype, mode --)
28	VRQ_LOCATOR (N/A)	(x y xout yout term --)
31	VRQ_STRING (N/A)	(maxlength echo-mode echoxy string --)

devicetype: the device desired to be selected
1 = Locator (mouse)
2 = Valuator (up and down arrows)
3 = Choice (keyboard)
4 = String (string input)

mode: the mode desired to be selected
1 = request (wait until input)
2 = sample (do not wait for input)

x and y: the starting x and y coordinates for the mouse cursor (pointer).

xout: user-initialized variable that contains the x position of the mouse after termination of VRQ_LOCATOR.

yout: user-initialized variable that contains the y position of the mouse after termination of VRQ_LOCATOR.

term: user-initialized variable that contains the terminating code after termination of VRQ_LOCATOR.
30 = left button termination
31 = right button termination
other = ASCII equivalent from keyboard

maxlength: maximum specified length in characters of the input string.

echomode: determines whether or not keyboard input for VRQ_STRING is echoed to the device handle (but doesn't seem to work!)
0 = echo off
1 = echo on (doesn't work...)

echoxy: address of array containing word values of the x and y coordinates for the start of echoing. Needed even when echomode is set to "off."

string: user created array that will pop its address onto the stack. After VRQ_STRING has finished, the variable string will contain a C string that can be "packed" into a Forth string and then used by Forthmacs.
Note: these two input functions WILL NOT work with 4x FORTH.

EXAMPLE 4 (Forthmacs)

```
Type:
\ Move mouse and then terminate with button or
keypress
variable xout variable yout variable term
: waitlmouseymvt
   200000 0 do loop
   1 1 1 vsin_mode
   1 320 200 xy.scale xout yout term vrq_locator
   ." Starting point is center of screen, or" cr
   ." (320, 200) for mono," cr
   ." (320, 100) for medium res," cr
   ." (160, 100) for low res." cr
   ." End point is " xout ? ." , " yout ? cr
   ." Terminating action is = " term ?
;
waitlmouseymvt
```

EXAMPLE 5 (Forthmacs)

```
Type:
\ Request a string input
.( Please type something up to 20 characters long.)
1 4 1 vsin_mode
create astring 20 allot
create xyarray 0 w, 0 w,
1 15 0 xyarray astring vrq_string
.( You just typed) cr
\ creating another string (FORTH packed string)
create prstring 15 allot
\ convert C string to Forth string
astring astring cstrlen prstring pack drop
\ Print the string
prstring ".
```

Example 4 puts the mouse cursor on the screen and then lets the user move it around until a key or button is pressed. Once this happens, the word waitlmouseymvt (or WAIT until MOUSEY MoVemenT) will print out the ending coordinate of the mouse and the terminating value. The next example simply asks for a string, converts it into a Forth string, and prints the string out. If there is enough interest, I will write a column comparing Forth and C strings in the future; for now just copy the above to suit your needs.

Inquires: The Reversed Attributes

VDI inquiries are basically calls that return the values of the VDI attributes such as line types, line color, fill types, text sizes, etc. The general form for a VDI inquiry is ([some parameters] array --). The parameters may vary, but the address of the array must be passed so that the various attribute values can be inserted into the array. The array should be anywhere from 6 to 64 bytes long and will almost always be segmented by words (16 bits) instead of the normal 32 bit divisions. The most difficult aspect of understanding inquiries is the interpretation of what the numbers in the array mean. Once there is a reference source for these numbers, using inquiries becomes easy.
Table 4: INQUIRIES
Note: To understand the ranges and meaning of parameters further, refer to the September '86 A Step Forth and the above discussion of text attributes.

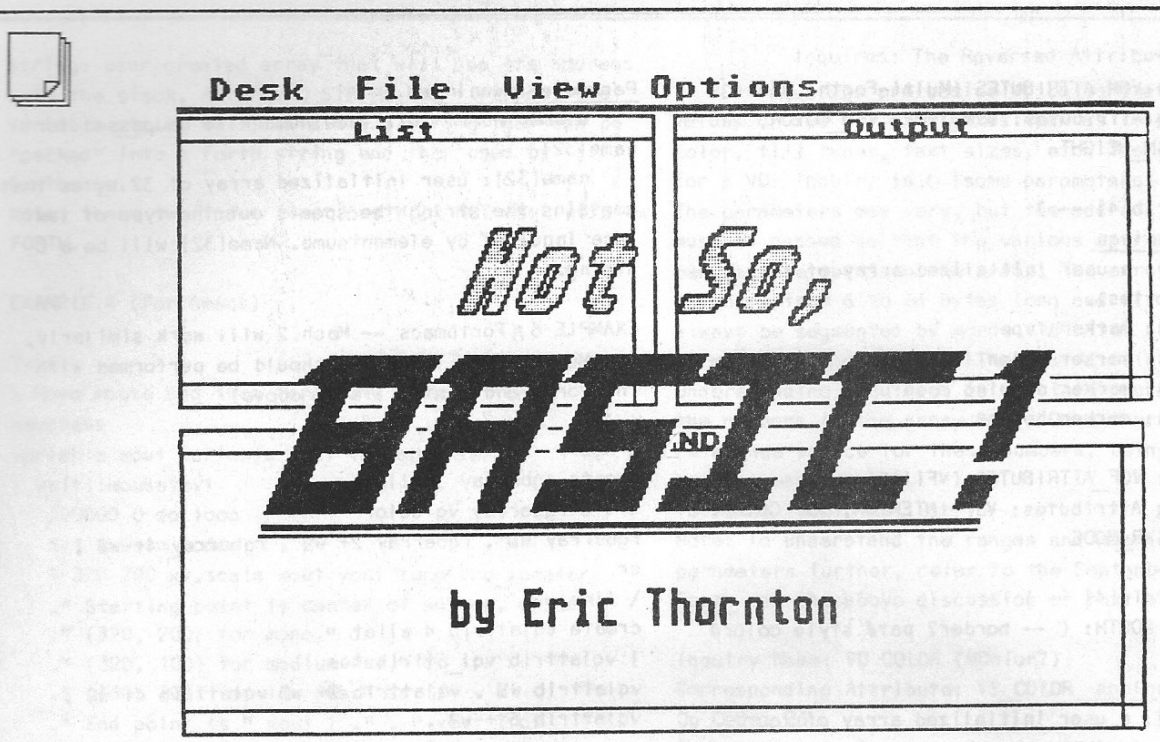
Inquiry Name: VQ_COLOR (VColor?)
Corresponding Attribute: VS_COLOR
Op Code: 26
Syntax: (colorindex setflag rgb[3] --)
Multi-FORTH: (colorindex setflag -- red green blue colorindex)
Parameter meanings
colorindex: the # of the color to be inquired upon (0-15 low res, 0-3 medium res, 0-1 high res)
setflag: 0 = return index of indexed color (will try find color make up of index 13 in medium res, for example.)
1 = return index of realized color (will adjust if index is out of range)
rgb[3]: a user initialized array of three elements (6 bytes).
rgb(1) (the first element of the array rgb[3]) or red: red intensity
rgb(2) (the second element of array rgb[3] or green: green intensity
rgb(3) or blue: blue intensity

Inquiry Name: VQL_ATTRIBUTES (VLine?)
Corresponding Attributes: VSL_TYPE, VSL_COLOR, VSWR_MODE, VSL_WIDTH
Op. Code: 35
Syntax: (attrib[4] --)
Multi-FORTH: (-- width style color# draw-mode)
Parameter meanings
attrib[4]: a user initialized array of four elements (8 bytes).
attrib(1) (first element of the array attrib[4]) or style: line type
attrib(2) or color#: line color index
attrib(3) or draw-mode: line drawing mode
attrib(4) or width: line width

```
Inquiry Name: VQM_ATTRIBUTES (Multi-Forth: N/A)
Corresponding Attributes: VSM_TYPE, VSM_COLOR,
VSWR_MODE, VSM_HEIGHT
Op. Code: 36
Syntax ( attrib[4] -- )
Parameter meanings
    attrib[4]: a user initialized array of four
elements (8 bytes).
    attrib(1): marker type
    attrib(2): marker color index
    attrib(3): marker drawing mode
    attrib(4): marker height

Inquiry Name: VQF_ATTRIBUTES (VFill?)
Corresponding Attributes: VSF_INTERIOR, VSF_COLOR,
VSF_STYLE, VSWR_MODE
Op. Code: 37
Syntax: ( attrib[4] -- )
      Multi-FORTH: ( -- border? pat# style color#
draw-mode )
Parameter meanings
    attrib[4]: a user initialized array of four
elements (8 bytes).
    attrib(1) or style: interior fill style
    attrib(2) or color#: interior fill color index
    attrib(3) or pat#: interior fill pattern
    attrib(4) or draw-mode: interior fill drawing
mode
    border?: border on/off

Inquiry Name: VQT_ATTRIBUTES (VText?)
Corresponding Attributes: VST_FONT, VST_COLOR,
VST_ROTATION, VST_ALIGNMENT, VSWR_MODE, VST_POINT,
VST_HEIGHT
Op. Code: 38
Syntax: ( attrib[10] -- )
      Multi-FORTH: ( -- angle horz vert font# color#
draw-mode )
Parameter meanings
    attrib[10]: a user initialized array of ten
elements (20 bytes).
    attrib(1)or font#: text face
    attrib(2) or color#: text color index
    attrib(3) or angle: text rotation angle
    attrib(4) or horz: horizontal text alignment
    attrib(5) or vert: vertical text alignment
    attrib(6) or draw-mode: text writing mode
    attrib(7): character width
    attrib(8): character height
    attrib(9): character cell width
    attrib(10): character cell height

Inquiry Name: VQT_NAME (Multi-FORTH: N/A)
Op. Code: 130
Syntax: ( elementnumb name[32] -- )
Parameter meanings
    elementnumb: text face number to be passed for
name[32]
    name[32]: user initialized array of 32 bytes that
contains the string the speels out the type of text
face inquired by elementnumb. Name[32] will be a C
string.

EXAMPLE 6 (Forthmacs -- Mach 2 will work similarly,
and Multi-FORTH inquiries should be performed with
the words and syntax listed above)

\ Rgb
create rgbarray 3 allot
1 1 0 rgbarray vq_color
rgbarray w@ . rgbarray 2+ w@ . rgbarray 4+ w@ .
cr
\ Line
create vqlattrib 4 allot
1 vqlattrib vql_attributes
vqlattrib w@ . vqlattrib 2+ w@ vqlattrib 4+ w@ .
vqlattrib 6 + w@ .
cr
\ Fill
create vqfattrib 4 allot
1 vqfattrib vqf_attributes
vqfattrib w@ . vqfattrib 2+ w@ vqfattrib 4+ w@ .
vqfattrib 6 + w@ .
cr
\ Text
create vqtattrib 10 allot
1 vqtattrib vqt_attributes
vqtattrib w@ . vqtattrib 2+ w@ vqtattrib 4+ w@ .
vqtattrib 6 + w@ .
vqtattrib 8 + w@ . vqtattrib 10 + w@ vqtattrib 12 +
w@ . vqtattrib 14 + w@
vqtattrib 16 + w@ . vqtattrib 18 + w@ .
cr
\ Text face
create name 32 allot
1 0 name vqt_name
: saytext 32 0 do name i + c@ emit loop ;
saytext cr
variable inputmode
1 1 inputmode vqin_mode
\ Input mode =
inputmode @ .
```

The above example shows how to get the values of inquiries. There are still more VDI functions, and perhaps if there is enough demand, I can discuss VDI at an even greater length. Next month, I will present the different ways AES is accessed by various ST Forth systems. Some methods are simple and painless, and others are not so easy. Each offers its own

Continued on page 47 -->



A highly placed source at ATARI informed me a couple of months ago that the new ST BASIC was in the last debug stages, and will remain so for an indeterminate length of time, (which should not exceed 6 months), and that we should not be displeased with the finished product. Well, depending on how I feel at the time I could be displeased with Ghandi.

If neccessary I will also support in this column other BASICs which are already on the market.

To review slightly (for those who may have missed the last issue.) I am covering the more likely VDI and AES calls from ST BASIC.

Now before we go into a full scale application in ST BASIC, let's go over the full calling proceedure for VDI and AES calls. If you missed the last issue, Beg, Borrow or Buy a copy.

VDI OPERATIONS (Only common ones)

Common variables

?= read description of function for value to place here.

X1= x coordinate of first point
Y1= y coordinate of first point
X2= x coordinate of second point
Y2= y coordinate of second point
...= Repeat same type of values.

VDI ATTRIBUTE FUNCTIONS...continued

SET GRAPHIC TEXT COLOR:

Lets you set the color which will be used for all graphic text, until changed.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	22	reg no.			
+2	0				
+4					
+6	1				

SET TEXT EFFECTS:

The effect is set by the 6 most least significant bits passed in intin+0. Add up these values as required. 0=normal, 1=bold, 2=light, 4=italic, 8=underline, 16=outline, 32=shadowed.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	106	effect			
+2	0				
+4					
+6	1				

FILL FUNCTIONS:

The normal ST BASIC commands for color and fill pattern do the same job as the equivalent VDI calls. So use BASIC's commands instead.

SET FILL PERIMETER VISABILITY:

Not supported by BASIC is fill perimeter. This

is almost (but not quite) as easy as
S calls are made in a slightly
er than VDI calls. First the arrays
command are not as clear as the VDI
tarting addresses for the GEMSYS
d in memory in a table called GB, and
u get them out.
slightly different names for the
to differentiate them from the VDI
milar functions. The array CONTROL,
ke CONTRL, in this case it is handled
command and you won't need to worry

+8)*2^16+peek(gb+10)
' funny but, what it does is: Peek a
ight bytes from the start of table GB
hat so it becomes the hi word of this
retrieving and then peek the low word
further along in this table and add

and col=peek(intout+2). Val=1 if point set.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	105		x1	val	
+2	1		y1	col	
+4					
+6	0				

VDI INPUT FUNCTIONS:

These functions are primarily not useful, unless you are going to throw out ST BASIC's entire environment. Which we will do later, hence these fuction will be explained at that time.

GET MOUSE BUTTONS AND LOCATION:

Pretty straight-forward, poke in the right values, make the call, peek out the answer. This function from BASIC is so slow... I can't recommend that you use it for anything which requires a quick response time. Bval=1 if button pressed. X1 and Y1 are the x and y coordinates of the current location.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	124			bval	
+2	0				x1
+4					y1
+6	0				

AES CALLS

Application Environment Services: It may sound like a social service orgranization, but it isn't.

Most of the AES calls are out of the range of most BASIC programs, so I'll just cover a few of the more important and useful ones.

Most of the time this is the only array you will need to know about for GEMSYS calls. In case you're curious here are the formuli for the rest of them.

```
GINTOUT=peek(gb+12)*2^16+peek(gb+14)
GADDRIN=peek(gb+16)*2^16+peek(gb+18)
GADDROUT=peek(gb+20)*2^16+peek(gb+22)
```

Here is an example which shows how to use GEMSYS to find out the graphics handle of this application (BASIC) and the current width and height of characters and their rows and columns.

```
10 REM call GEMSYS (graf_handle)
20 GINTOUT=peek(gb+12)*2^16+peek(gb+14)
30 GEMSYS 77            :' 77 is the opcode for this
function
40 grhandle=peek(GINTOUT+0)   :' graphics handle
50 charw=peek(GINTOUT+2)
60 charh=peek(GINTOUT+4)
70 colw=peek(GINTOUT+6)
80 rowh=peek(GINTOUT+8)
```

GET INFORMATION ABOUT A WINDOW: GEMSYS 104

Peek the AES window handle from systab+? (see BASIC SOURCEBOOK) and use it here to find out about the window. Inner, outer, previous, max sizes of the window, positions of the slider, etc. For now I'll show you how to use it to find the inner dimentions of a window in pixels (probably the most useful).

More -->

User Group Special

Just send in 5 or more subscriptions at the same time together with a letter or newsletter from your ST-oriented User Group or ST Special Interest Group (SIG) and qualify for a yearly subscription to ST Applications for only $30 each. That's right, 12 issues for $2.50 each!!

There's no limit to how many members can apply as long as 5 or more subscribe at a time.

ST Applications
THE ATARI 520/1040ST JOURNAL

P.O. Box 980
Forestville, CA 95436

A highly placed source at ATARI informed me a couple of months ago that the new ST BASIC was in the last debug stages, and will remain so for an indeterminate length of time, (which should not exceed 6 months), and that we should not be displeased with the finished product. Well, depending on how I feel at the time I could be displeased with Ghandi.

If neccessary I will also support in this column other BASICs which are already on the market.

To review slightly (for those who may have missed the last issue.) I am covering the more likely VDI and AES calls from ST BASIC.

Now before we go into a full scale application in ST BASIC, let's go over the full calling proceedure for VDI and AES calls. If you missed the last issue, Beg, Borrow or Buy a copy.

VDI OPERATIONS (Only common ones)

Common variables

?= read description of function for value to place here.

X1= x coordinate of first point

Y1= y coordinate of first point

X2= x coordinate of second point

Y2= y coordinate of second point

...= Repeat same type of values.

VDI ATTRIBUTE FUNCTIONS...continued

SET GRAPHIC TEXT COLOR:

Lets you set the color which will be used for all graphic text, until changed.

Element	Contrl	Intin	Ptsin	Intout	Ptsout
+0	22	reg no.			
+2	0				
+4					
+6	1				

SET TEXT EFFECTS:

The effect is set by the 6 most least significant bits passed in intin+0. Add up these values as required. 0=normal, 1=bold, 2=light, 4=italic, 8=underline, 16=outline, 32=shadowed.

Element	Contrl	Intin	Ptsin	Intout	Ptsout
+0	106	effect			
+2	0				
+4					
+6	1				

FILL FUNCTIONS:

The normal ST BASIC commands for color and fill pattern do the same job as the equivalent VDI calls. So use BASIC's commands instead.

SET FILL PERIMETER VISABILITY:

Not supported by BASIC is fill perimeter. This

function lets you turn on and off the visability of the perimeter on fill functions. Flag value=0 if frame to be turned off, and 1 if frame on wanted.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	104	flag			
+2	0				
+4					
+6	1				

RASTER OPERATIONS:

This set of essentially interesting but looney functions are practically impossible to explain. However, I will explain them in a later issue. (when I have more space.) They can be used from basic to BLIT a section of screen around or to copy images from array to screen and back again.

GET PIXEL:

Returns the set or cleared status and the color of any point on the screen, in val=peek(intout+0) and col=peek(intout+2). Val=1 if point set.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	105		x1	val	
+2	1		y1	col	
+4					
+6	0				

VDI INPUT FUNCTIONS:

These functions are primarily not useful, unless you are going to throw out ST BASIC's entire environment. Which we will do later, hence these fuction will be explained at that time.

GET MOUSE BUTTONS AND LOCATION:

Pretty straight-forward, poke in the right values, make the call, peek out the answer. This function from BASIC is so slow... I can't recommend that you use it for anything which requires a quick response time. Bval=1 if button pressed. X1 and Y1 are the x and y coordinates of the current location.

Element -	Contrl	Intin	Ptsin	Intout	Ptsout
+0	124			bval	
+2	0				x1
+4					y1
+6	0				

AES CALLS

Application Environment Services: It may sound like a social service orgranization, but it isn't.

Most of the AES calls are out of the range of most BASIC programs, so I'll just cover a few of the more important and useful ones.

Calling AES is almost (but not quite) as easy as VDI calls. AES calls are made in a slightly different manner than VDI calls. First the arrays for the GEMSYS command are not as clear as the VDI arrays. The starting addresses for the GEMSYS arrays are held in memory in a table called GB, and this is how you get them out.

I will use slightly different names for the GEMSYS arrays to differentiate them from the VDI arrays with similar functions. The array CONTROL, while it is like CONTRL, in this case it is handled by the GEMSYS command and you won't need to worry about it.

```
GINTIN=peek(gb+8)*2^16+peek(gb+10)
```

Looks kinda' funny but, what it does is: Peek a word that is eight bytes from the start of table GB and multiply that so it becomes the hi word of this address we are retrieving and then peek the low word from two bytes further along in this table and add it in.

Most of the time this is the only array you will need to know about for GEMSYS calls. In case you're curious here are the formuli for the rest of them.

```
GINTOUT=peek(gb+12)*2^16+peek(gb+14)
GADDRIN=peek(gb+16)*2^16+peek(gb+18)
GADDROUT=peek(gb+20)*2^16+peek(gb+22)
```

Here is an example which shows how to use GEMSYS to find out the graphics handle of this application (BASIC) and the current width and height of characters and their rows and columns.

```
10 REM call GEMSYS (graf_handle)
20 GINTOUT=peek(gb+12)*2^16+peek(gb+14)
30 GEMSYS 77          :' 77 is the opcode for this
function
40 grhandle=peek(GINTOUT+0)   :' graphics handle
50 charw=peek(GINTOUT+2)
60 charh=peek(GINTOUT+4)
70 colw=peek(GINTOUT+6)
80 rowh=peek(GINTOUT+8)
```

GET INFORMATION ABOUT A WINDOW: GEMSYS 104

Peek the AES window handle from systab+? (see BASIC SOURCEBOOK) and use it here to find out about the window. Inner, outer, previous, max sizes of the window, positions of the slider, etc. For now I'll show you how to use it to find the inner dimentions of a window in pixels (probably the most useful).

More -->

Element -	GINTIN	GINTOUT	GADDRIN	GADDROUT
+0	wihand			
+2	4	x1		
+4		y1		
+6		width		
+8		height		

The rest of the AES window functions will have to wait for later as they are soooo very complex.

G1 GRAPHIC RUBBER BOX: GEMSYS 70

Wait until the left mouse button is pressed before calling this function, as it returns automatically when the button is released; immediately if it is not pressed at all. Same as the rubber box on the desktop. Passed are the x and y of the upper left of box and the minimum width and height the box may attain. Returned are the last width and height of the rubber box when the user let go of the left button.

Element -	GINTIN	GINTOUT	GADDRIN	GADDROUT
+0	x1			
+2	y1	lastw		
+4	miniw	lasth		
+6	minih			
+8				

GRAF MOUSE: GEMSYS 78

Change the mouse's current form to another predefined form. Values range from 0-7 and forms range from the arrow with which we are all familiar to an outlined crosshair.

Element -	GINTIN	GINTOUT	GADDRIN	GADDROUT
+0	mform		0	
+2			0	
+4				
+6				
+8				

There are a lot more AES calls, however, most of which you'll never use from BASIC. The ones I've covered are only those that I have found an occasion to use in my own programs, which I'll share with you as time goes on.

Look forward to the upcomming issues, as I am indeed building up to something. Next issue: Load your NEOchrome picture files into ST BASIC with their proper colors.

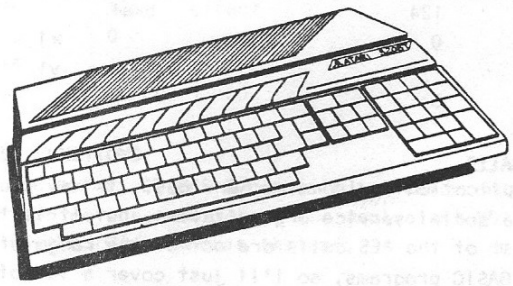
More Diskette Control from page 24

is one of the co-creators of the Miller code and is the creator of the "Miller-squared" code. Mr. Mallinson gave a lecture at a local university, on 1979Mar12, discussing "the development of digital tape recorders with extremely high data rates and areal densities.", and provided me with copies of three papers on magnetic recording. Both Mr. Mallinson and Mr. Miller were from Ampex at the time of this paper.

(2) "The Recording and Reproduction of Signals on Magnetic Medium Using Saturation-Type Recording" by J. J. Miyata and R. R. Hartel; IRE Transactions on Electronic Computers, Vol EC-8 No. 2, pp. 159:169, June 1959. "This paper discusses the various factors affecting the resolution in saturation magnetic recording. The effect on the recording process of the ... coating thickness, record-head gap width, head-to-coating separation, self-demagnetization, and record-head residual magnetization are discussed." This information is presented in tables and graphs, and equations are provided for calculating these effects.

(3) "Increased Magnetic Recording Read-back Resolution by Means of a Linear Passive Network" by H. M. Sierra; IBM Journal,pp. 22:33, January 1963. This paper describes a relatively complex passive network which narrows the data pulses in the playback amplifier in order to (a)improve data accuracy or (b) increase recording density. The network uses differential input and output, and uses only capacitors and inductors. Complete information is given in normalized form, and a specific example is also given.

There are many good papers on the subject of magnetic recording, but these three are the best overall. Copies can be obtained through the interLibrary Loan department of your local library. A College/University library may even have them (and other goodies!) in its collection.



First Machine Language Program

A Letter from Jim Wilson

by Tom McAllister

I have just finished your article in the August **ST APPLICATIONS**. I don't mean to be hard on you; God knows low-level programming is tough enough. But there are some problems with what you have written which are bound to get anybody following your path into problems.

The so-called documentation package is built around the idea that, as long as some basic rules are followed, it is possible to build libraries that can be called either from assembly language source code or from C source code (and presumably later from any other language which has a compiler/linker setup that conforms to the same standards).

With that in mind, the programmer should be encouraged to begin by writing all of his general assembly routines not into a program, but into libraries which can be linked to anything he writes later. Here are some of the standards:

1. Some registers cannot and others should not, be carried across library routines. Registers d0-d2 and a0-a2 are universally treated as scratch registers, and are not protected in library routines. If you pass control to a library routine and you are using any of these 6 registers, you must protect them before you pass control, and recover them on return, because whatever is in the standard library will NOT do it for you.

2. Register d0 is the 'message drop' register, even for functions that return nothing but error messages. If a routine returns a single value, it will be in d0. If there is an error, the error message is passed in d0. If a routine returns more than one value to a caller, then d0.l is a pointer to those values in memory.

3. Arguments passed to a library routine must be pushed, in reverse order, on the current stack, keeping in mind that byte size operations on the stack are illegal. Arguments may either be data (16, 32-bits), or pointers (32-bits), or a mix. 8-bit data must be right justified in 16-bit format, and forever treated as a word.

4. Library routines make liberal use of stack frames. The 68000 lacks a frame pointer, and register a6 is the universal choice. It may be best not to use a6 for any other purpose.

5. Each program may have one and only one starting point, which must be labeled 'main'. The starting point may be either in C ('main') or assembly ('_main') source code. The reason for the underscore is to make an assembly language source comply to common standards; the second phase of the C compiler appends an initial underscore, so to call any routine in a C compatible library from assembly, or to place an assembly language routine into a library callable from C, the programmer must add an underscore.

6. A library routine, and every label sought by the linker, including the assembly label '_main', must be declared as 'globl':

```
globl   _label1,label2
```

(Note that '_label1' could be called from either C or assembly, but 'label2' only from assembly, even if label2 conforms in all other respects to the library standards.) You can, for example, print out your null terminated message strings by simply linking to the '_printf' function in 'gemlib.o' and supported by libf.o:

```
        pea       string(pc)
        bsr       _printf
```

7. The support libraries included in the documentation package, and the ones to be written, work with the 'startup' object modules that are essential in linking together programs from libraries and source, such as 'apstart.o' and 'gemstart.o'.

Do not forget to begin your main assembly source code with:

```
        globl     _main
        _main     ...
```

to maintain 'startup' compatibility.

If the first column in a line of assembly is occupied by something other than an asterisk, it is assumed to be a label. Labels that begin in this column do not need a terminating colon:

```
label   move.l    (a0)+,(a1)+
        dbf       d0,label
```

Pseudo-ops do not have to begin with a dot:

```
        bss
```

but you can use dots if you like. The assembler will have no trouble at all with any mix of colons, no colons, dots, no dots, as long as you adhere to columnar rules. It is legal to indent labels and terminate them with colons; it may even be neater. But it is only a stylistic device, and not particularly easy to read.

It is WRONG, WRONG, WRONG to advocate departing from system programming standards. Either 'apstart.o' or 'gemstart.o' (there are other startups for GEM) do a lot of housekeeping, including stack management, as long as one or the other is declared to the linker, and with a '_main' in the primary source:

```
link68 [u] %1.68k=apstart,%1,libraryx,libraryz
```

The programmer can, and probably should, be encouraged to drop the 'text' pseudo-op, and refer to constant data and strings as:

```
        lea       label(pc)
```

This saves one word of code, and several cycles in execution; there is a problem, however, if the PC is > 32K from the target, due to wraparound. There are ways around that, too, but no problem until you write big programs.

Similarly, the programmer should be encouraged to reserve an address register as a static base pointer (a5 preferred), and load it with the address of the first item in the 'bss'. He can then code from that base pointer and avoid absolute addressing, which is C/PM86K's default (and very wasteful of memory and time) mode. The comments about wraparound apply here, too:

```
sb      equ       a5
        lea       label,sb
        move.w    label5(sb),d0
```

This will convert any address within 32K from the static base pointer into 'd16(sb)' form, again saving one word and several cycles per access.

There is an unfortunate misprint in Atari's documentation; the '-u' switch asks the assembler to reserve a space in the symbol table for all 'undefined (external)', not 'underlined', labels. Without this reservation, you couldn't link to libraries, because assembly would fail on the first undefined label. The linker resolves all external references, looking for matching 'globl' declarations in the libraries on the link line.

Hang in there! Nobody said it was supposed to be easy.

Jim Wilson
Los Angeles, CA

Response

I am impressed that you would put the effort you did into your letter commenting on my **First Machine Language Program.** Of course the points you make about observing conventions are crucial. I tried to link the starting subroutine in object code and failed as you know. You explain how. The advantage you point out is that the module might be invoked from C and assembly both. I intend to master the technique and appreciate the specifics in your letter. I also appreciate your concern for order and thank you for your patience. I want to follow the consensus on register allocation. Thanks for the information. The fact that this is no trivial matter and is a deliberate part of the ST system illustrates a point I myself was trying to make, namely machine language on ST goes far beyond the instruction set of the 68000. I am glad to see it all documented in **ST APPLICATIONS**, rather than each man eke it out on his own. More!

You are right in detecting that I don't warm to the idea of subroutines as linked object code. I like to see what I'm doing. Although far from

programming in C, I do want to understand how my doings stack up against the requirements of the Development C compiler and also the other C compilers, present and to come. I'd like to understand the non-C languages; not syntax, rather the impact of the compilers upon ST machine-programming. I wonder how Modula-2 reacts to '_main'? Assemblers appear to be proliferating, but will many have linkers?

I am heartily in favor of building library modules. Cannot make progress until I collect some (ever the bane of Assembler Programming). I may take advantage of certain C libraries, mainly floating point. Until Richard Frick of Atari straightened me out last week, I believed the documentation where it said no floating point was provided to add, subtract, multiply, divide.

Out of habit, I intended to disassemble to incorporate C Library modules. It dawned only slowly that I could link; which could prove a great boon, provided I don't have to disassemble in order to determine the parameters. You were good enough to mention general guidelines on parameters. Here then is what the conventions you advocate may do for me as a machine language programmer; give me easy access to C-libraries. ("Today the parameters, tomorrow the world.") I will label with '_main'; costs nothing and permits linking any time later at the binary level.

What I advocate departing from is not C. Frankly it is GEM and the VDI. I like GEM as an interface for the Disk Operating System but not universally in applications. The GEM/VDI 'system' seems impossible to avoid within the rules being followed to program ST without learning how the DOS functions. I seek instead to learn the DOS and to bypass the 'rules' in order to avoid GEM/VDI.

Programming with GEM and the VDI is highly complex. I am afraid ST will grow obsolete before I learn GEM in my spare time. Working outside the system may be no crime, I plead, under circumstances where I have little prospect of accommodating the system in my lifetime or the life of the computer. I shudder to think which. At the same time I hesitate to credit the wholesale programming efficiencies claimed for GEM/VDI, and I question the bid for universal applicability.

The tiny program in the article excerpted a goodly portion of the stack handling from the source equivalent 'apstart.s' of the linkable utility 'apstart.o'. Stack handling is an instance to me of DOS capabilities which really demand explicit exposition. Perhaps you agree. I've been experimenting with the Symbolic Interactive Disassembler, SID. You know, SID can choose and load another program above itself and tell where in memory the chosen landed. When I load different programs, their starting points, which I take to be SID-ending points, vary appreciably. SID in fact goes in and out like an accordion. Some fancy stack handling going on there; worthy of exposition.

Ahem! Did they get this fancy stack handling by linking a utility? I am ill-disposed to program by rote rules. If developing for commercial use, maybe... because there is nothing like a bottom line to concentrate attention upon rules. Yet I think of possibly uncovering unsuspected commercial advantage. I remember the unforeseen benefits reaped from gratuitous thoroughness while engineering. Isn't GEMDOS significantly different from CP/M-68K? Who is up on GEMDOS? I merely ask rhetorically.

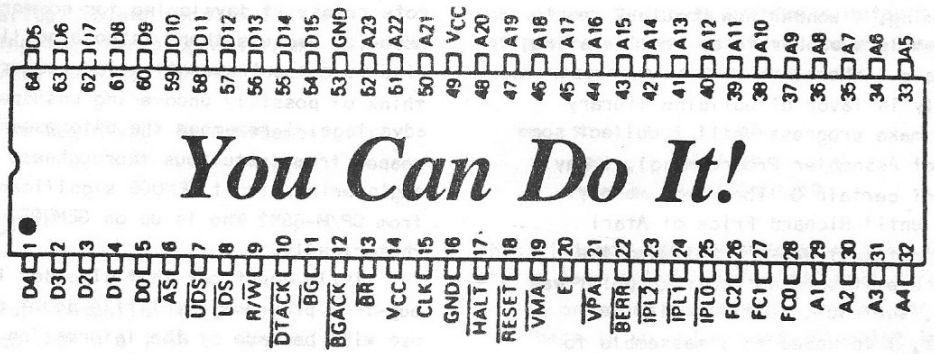
Little doubt but that disorder is enemy of bug-free programming. Little doubt on that. Good use will be made of the information and your advice. The part about commanding the assembler needs time to be assimilated. In effect, I plead guilty with an explanation. No one did tell me it would be easy but imagine if we could make it easy: A new ST landmark program would appear every week and with unmatched display, ST would emerge unbeatable!

Tom Mc Allister

Moving?

We all make mistakes.
If you have a subscription problem please call
(707) 869-2325
or
write us at
ST Applications
P.O. Box 980
Forestville, CA 95436
If possible, please have your mailing label available, as well as your canceled check if you are having problems with payment.
If moving, please give us both your old address and new address.
Thank You!

Subscription Problem?



Description of MC68000 pins

by David Bessey

This month: a brief description of the functions of the MC68000's pins. When the discussion turns to actual hardware applications I will make timing diagrams to show how the signals are used in real life. The nDTACK signal, for instance, is supplied by the user's hardware, but it controls two or more MPU states.

†A00:†A23 Address lines

nAS Address Strobe; indicates that the address lines are valid.

nBERR Bus ERRor; this input tells the MPU that there is a 'fatal' bus error and allows recovery from the error

nBG Bus Grant; indicates that the MPU will relinquish control of the bus at the end of the current bus cycle.

nBGACK Bus Grant Acknowledge; indicates that some other device has actually taken control of the bus.

nBR Bus Request; indicates that some other device wants control of the bus.

CLK CLocK; 8MHz input; 50% duty cycle.

†D00:†D0F Data lines

nLDS Lower Data Strobe; indicates that data bits 00:07 are valid.

nUDS Upper Data Strobe; indicates that data bits 08:0F are valid.

nDTACK Data Transfer ACKnowledge; special handshake signal required for data transfer.

E Enable/clock; E clock output for 6800 series peripheral devices. 40%/60% duty cycle derived from the 8MHz CLK input.

†FC0:†FC2 Function Code; these lines indicate which address space is being used. (described in Vol.1 #1)

GND GrouND; these pins are power supply ground lines (zero volts,Vss).

nHALT HALT; input: halts the MPU.
output: indicates MPU is halted.

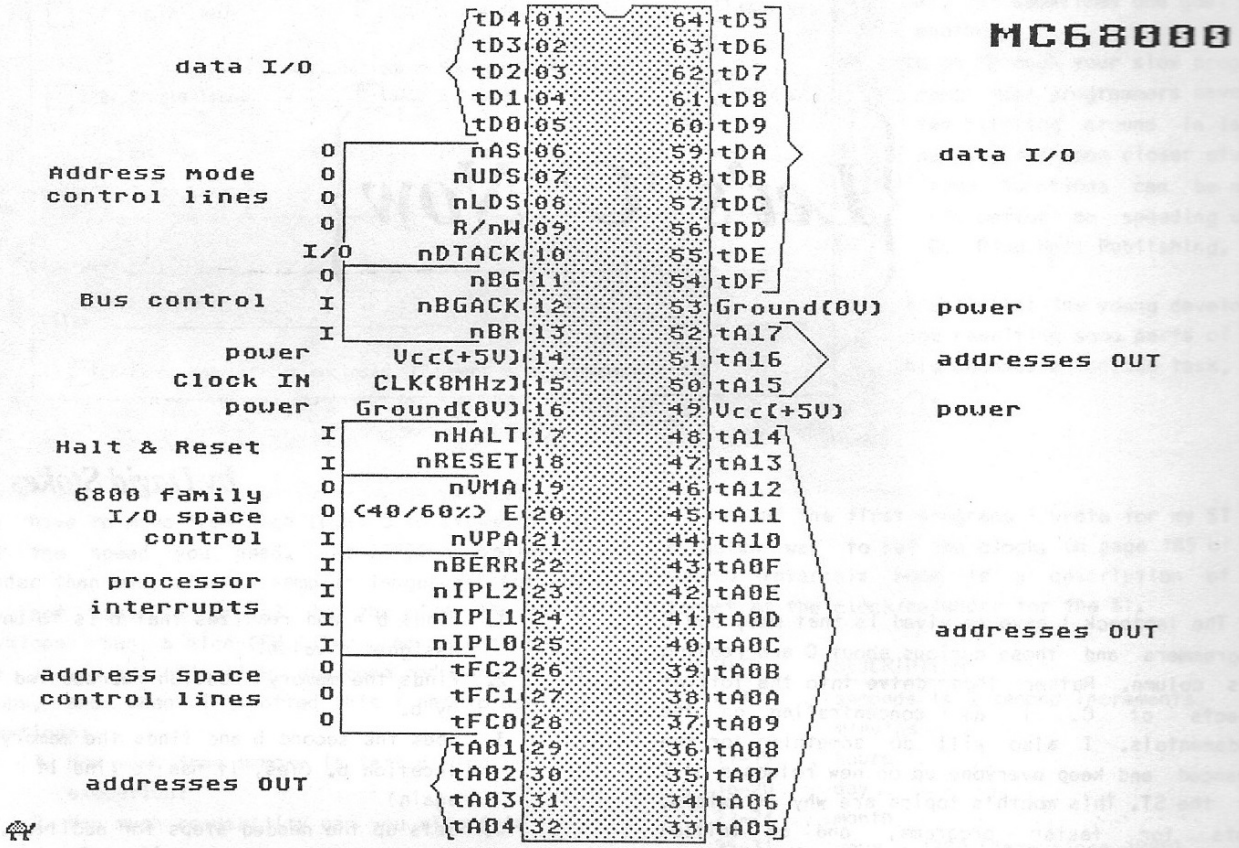


Figure 1. Pinouts for the Motorola 68000 Microprocessor.

---

nIPL0:nIPL2
: Interrupt Priority Level; if the interrupt number on these pins is greater-than-or-equal-to the interrupt mask in the status register, an interrupt exception occurs.

tR/nW
: Read/notWrite; high: data transferred into MPU low: data transferred from MPU

nRESET
: input : resets MPU output: resets peripheral devices.

Vcc
: these pins are the +5V power input.

nVPA
: Valid Peripheral Address; this input indicates that the MPU is addressing an M6800-series peripheral device, and that data transfer will be synchronized by the E clock. Automatic vectoring is used for interrupts.

nVMA
: Valid Memory Address; This output operates only when nVPA is low. It tells the 6800 peripherals that (a) their address lines are valid, and (b) data transfer is synchronized to the E clock.

{Let's 'C' Now}

by David Stokes

The feedback I have received is that many new C programmers and those curious about C are reading this column. Rather than delve into the loftier aspects of C, I am concentrating on the fundamentals. I also will do something for the advanced and keep everyone up on new releases of C for the ST. This month's topics are why ++ exists, hints for faster programs, and an initial impression of **Mark Williams C.**

Hints for C novices

The C language is very similar to English. Both languages have nuances that will drive novices up the wall. C often looks stranger than the same program written in PASCAL, Modula-2, or any of the popular high level languages available for the ST.

On the other hand, BASIC is very easy to understand for almost any person starting to program a computer. BASIC makes simple algebraic statements like b = b + 1. This makes sense to many people.

C was designed as a way to avoid writing in assembly language while retaining the power of assembler. People who write in assembler or C value short, compact program code because it runs faster. So b = b + 1 becomes b++ (++ increments the value of the variable by one).

Why did this shorthand develop? To make the resulting programs run faster. When the compiler reads a program and sees b = b + 1, it performs the following steps:

1. Finds b = and realizes that b is to be assigned a value.
2. Finds the memory location represented by b.
3. Sees the second b and finds the memory location b. (Yes, it has to find it again)
4. Sets up the needed steps for addition.
5. Takes 1 as the item to be added to b.
6. Performs the addition b + 1.
7. Puts the result in the memory location b.

For b++, the steps are:

1. Finds b and fetches the value of that location in memory.
2. Increments b.
3. Replaces the value in memory.

I have greatly simplified the process of compiler parsing, but you should see the idea. Many people argue that PASCAL, BASIC, Modula-2, ADA, etc., are easier to read than C programs filled with ++'s and --'s. However, a little time spent learning the unique features of C will be rewarded with faster running programs.

At first the ++, ?:, --, &, and etc. are confusing but 99.99% of all C programmers keep a cheat sheet near their keyboard. The proper name of this cheat sheet is the Index to C operators. The idea is similar to math rules like 'multiplication is done before addition'.

If you wish to delve into C and have no previous experiences, it is suggested that you:

1. Buy a copy of a C compiler. Read the end of

ST Applications
THE ATARI 520/1040ST JOURNAL

P.O. Box 980
Forestville, CA 95436

☐ $5/ Single issue ☐ $20/ 6 Issues ☐ $40/ yr.

====> ST Applications + Support Disk <=====

☐ $12/ Single Issue ☐ $50/ 6 Issues ☐ $100/ yr.

☐ Renewal Start Subscription with _____ issue.

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

☐ Check or Money Order enclosed (Payment must accompany order)

ave self documenting code (lots
who would comment 1800 lines of
en the need to read a program
to unroll. Readability of your
oal, but sometimes one goal has
another.
to go through your slow program
comb! Most programmers have an
two sitting around in large
no use. And upon closer study,
some functions can be made
information on speeding up C
C, Plum Hall Publishing, by

t's limits! The young developer
now rewriting some parts of his
his becomes a tedious task, but
ıry.

faster that the other high level languages, but you may have to move from high level C to assembler to get the speed you need. The 68000 assembler is kinder than many other assembler languages, but you may not want to grovel in the bits. The young developer had a nice GEM interfaced spread sheet, but it was slightly slow. His code had many nested loops, and when I spotted this I had to ask two questions:

1. How much free memory is left during execution?
2. How much readability can you afford to give up?

The first question is based on the knowledge that a branching instruction takes the computer more time than an assignment. The overhead in jumping around memory can bog down a program. So we unrolled a few loops. Unrolling a loop is basically like so:

```
for ( i = 0 ; i < 5 ; i++)
     foo[i] = NULL;
```

becomes

```
foo[0] = NULL;  foo[1] = NULL;  foo[2] = NULL;
foo[3] = NULL;  foo[4] = NULL;  foo[5] = NULL;
```

This saves milliseconds! Milliseconds are small, but this guy was looping so much that they began to add up to seconds! If you investigate the assembler code produced from the unrolled code it is easy to see the overhead necessary to loop. The trade-off is that by unrolling loops you take up more memory.

Nested loops can become extremely messy, and tracing a misplaced subscript is a real pain! It is

One of the first programs I wrote for my ST was a quick way to set the clock. On page 185 of the Abacus Internals book is a description of the format of the clock/calendar for the ST.

BITS	DESCRIPTION
0-4	seconds in 2 second increments
5-10	minutes
11-15	hours
16-20	day
21-24	month
25-31	year - 1980 (date since 1980)

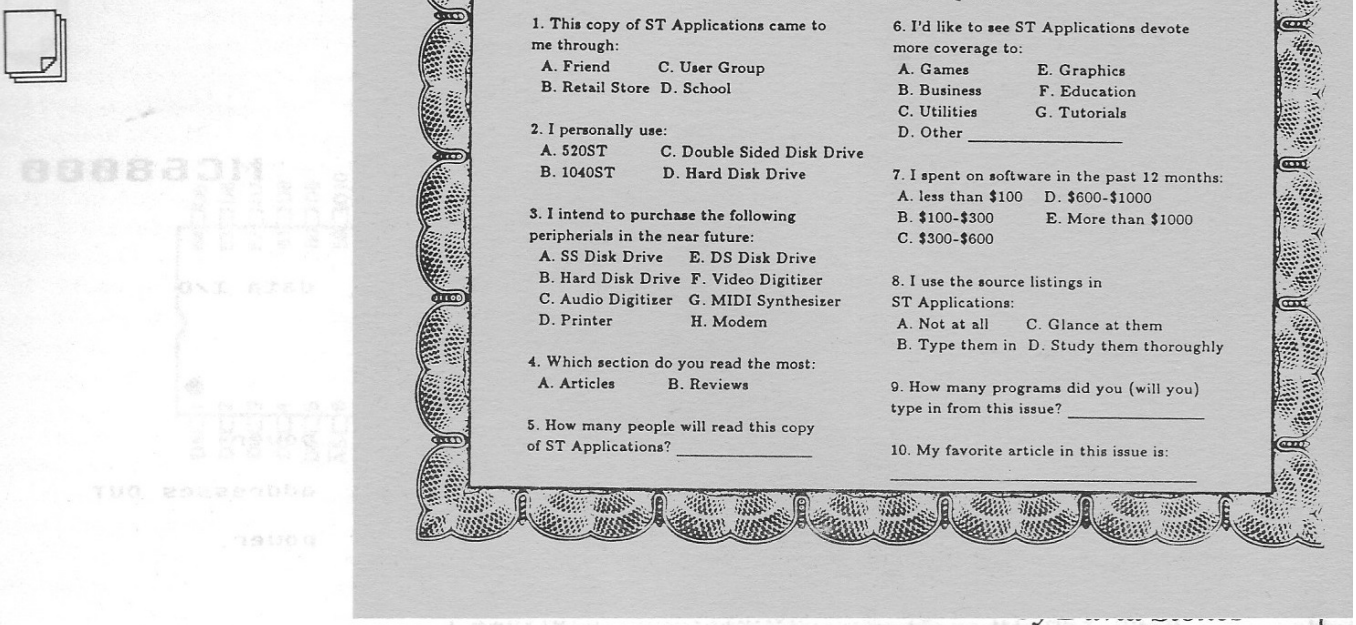
The trick to moving around the bits needed without getting the bits we don't want is shifting and masking. Shifting is sliding the bits around and masking is isolating bits. This month's example program, **exdate.c**, is a clear example of this S&M. Here is the output from a C-shell-like environment:

```
$date     Sat Aug 16 14:58:14 1986
$exdate   ST time D106F4E is 8/16/1986 at
14:58
```

The D106F4E is the system date/time format printed in hexadecimal. Breaking down the D106F4E into binary we get:

```
11010001000001101111010011110
```

To get seconds we AND (&) the time with 1F. We need to screen five bits to get the seconds and the hex code for five bits on is 1F (11111 binary). The math looks like:



Please Answer Each of These Questions

1. This copy of ST Applications came to me through:
   A. Friend C. User Group
   B. Retail Store D. School

2. I personally use:
   A. 520ST C. Double Sided Disk Drive
   B. 1040ST D. Hard Disk Drive

3. I intend to purchase the following peripherials in the near future:
   A. SS Disk Drive E. DS Disk Drive
   B. Hard Disk Drive F. Video Digitizer
   C. Audio Digitizer G. MIDI Synthesizer
   D. Printer H. Modem

4. Which section do you read the most:
   A. Articles B. Reviews

5. How many people will read this copy of ST Applications? _____

6. I'd like to see ST Applications devote more coverage to:
   A. Games E. Graphics
   B. Business F. Education
   C. Utilities G. Tutorials
   D. Other _____

7. I spent on software in the past 12 months:
   A. less than $100 D. $600-$1000
   B. $100-$300 E. More than $1000
   C. $300-$600

8. I use the source listings in ST Applications:
   A. Not at all C. Glance at them
   B. Type them in D. Study them thoroughly

9. How many programs did you (will you) type in from this issue? _____

10. My favorite article in this issue is:
    _____

The feedback I have received is that many new C programmers and those curious about C are reading this column. Rather than delve into the loftier aspects of C, I am concentrating on the fundamentals. I also will do something for the advanced and keep everyone up on new releases of C for the ST. This month's topics are why ++ exists, hints for faster programs, and an initial impression of **Mark Williams C.**

Hints for C novices

The C language is very similar to English. Both languages have nuances that will drive novices up the wall. C often looks stranger than the same program written in PASCAL, Modula-2, or any of the popular high level languages available for the ST.

On the other hand, BASIC is very easy to understand for almost any person starting to program a computer. BASIC makes simple algebraic statements like b = b + 1. This makes sense to many people.

C was designed as a way to avoid writing in assembly language while retaining the power of assembler. People who write in assembler or C value short, compact program code because it runs faster. So b = b + 1 becomes b++ (++ increments the value of the variable by one).

Why did this shorthand develop? To make the resulting programs run faster. When the compiler reads a program and sees b = b + 1, it performs the following steps:

1. Finds b = and realizes that b is to be assigned a value.
2. Finds the memory location represented by b.
3. Sees the second b and finds the memory location b. (Yes, it has to find it again)
4. Sets up the needed steps for addition.
5. Takes 1 as the item to be added to b.
6. Performs the addition b + 1.
7. Puts the result in the memory location b.

For b++, the steps are:

1. Finds b and fetches the value of that location in memory.
2. Increments b.
3. Replaces the value in memory.

I have greatly simplified the process of compiler parsing, but you should see the idea. Many people argue that PASCAL, BASIC, Modula-2, ADA, etc., are easier to read than C programs filled with ++'s and --'s. However, a little time spent learning the unique features of C will be rewarded with faster running programs.

At first the ++, ?:, --, &, and etc. are confusing but 99.99% of all C programmers keep a cheat sheet near their keyboard. The proper name of this cheat sheet is the Index to C operators. The idea is similar to math rules like 'multiplication is done before addition'.

If you wish to delve into C and have no previous experiences, it is suggested that you:

1. Buy a copy of a C compiler. Read the end of

this column for announcements on C compilers.

2. Go to the bookstore and get at least two different C language introductory texts. By referencing at least two books you'll be able to get a different view of an item that one author may poorly describe.

3. Read as many C programs as you can. Imagine a child learning to read but who only reads what he or she writes -- the kid goes no place fast. Do not hesitate to modify someone else's code (after you copy the original first!) and experiment.

C is not as hard as it first looks. Good luck.

Developers

It is safe to say that most of the software for the ST is written in C. But what happens if you develop your program, and it runs slowly? Recently I received a call from a developer asking if I could help him speed up his program. C is usually faster that the other high level languages, but you may have to move from high level C to assembler to get the speed you need. The 68000 assembler is kinder than many other assembler languages, but you may not want to grovel in the bits. The young developer had a nice GEM interfaced spread sheet, but it was slightly slow. His code had many nested loops, and when I spotted this I had to ask two questions:

1. How much free memory is left during execution?
2. How much readability can you afford to give up?

The first question is based on the knowledge that a branching instruction takes the computer more time than an assignment. The overhead in jumping around memory can bog down a program. So we unrolled a few loops. Unrolling a loop is basically like so:

```
for ( i = 0 ; i < 5 ; i++)
     foo[i] = NULL;
```

becomes

```
foo[0] = NULL;  foo[1] = NULL;  foo[2] = NULL;
foo[3] = NULL;  foo[4] = NULL;  foo[5] = NULL;
```

This saves milliseconds! Milliseconds are small, but this guy was looping so much that they began to add up to seconds! If you investigate the assembler code produced from the unrolled code it is easy to see the overhead necessary to loop. The trade-off is that by unrolling loops you take up more memory.

Nested loops can become extremely messy, and tracing a misplaced subscript is a real pain! It is very valuable to have self documenting code (lots of comments), but who would comment 1800 lines of foo[n] = 100? Often the need to read a program offsets the need to unroll. Readability of your program is a high goal, but sometimes one goal has to be sacrificed for another.

It also helps to go through your slow program with a fine tooth comb! Most programmers have an extra variable or two sitting around in large programs that have no use. And upon closer study, it is amazing how some functions can be made faster. (For more information on speeding up C programs: **Efficient C**, Plum Hall Publishing, by Plum and Brodie.)

But C does have it's limits! The young developer who called me is now rewriting some parts of his code in assembler. This becomes a tedious task, but sometimes is necessary.

Shifting and Masking

One of the first programs I wrote for my ST was a quick way to set the clock. On page 185 of the Abacus Internals book is a description of the format of the clock/calendar for the ST.

BITS	DESCRIPTION
0-4	seconds in 2 second increments
5-10	minutes
11-15	hours
16-20	day
21-24	month
25-31	year - 1980 (date since 1980)

The trick to moving around the bits needed without getting the bits we don't want is shifting and masking. Shifting is sliding the bits around and masking is isolating bits. This month's example program, **exdate.c**, is a clear example of this S&M. Here is the output from a C-shell-like environment:

```
$date     Sat Aug 16 14:58:14 1986
$exdate   ST time D106F4E is 8/16/1986 at
14:58
```

The D106F4E is the system date/time format printed in hexadecimal. Breaking down the D106F4E into binary we get:

```
11010001000001101111101001110
```

To get seconds we AND (&) the time with 1F. We need to screen five bits to get the seconds and the hex code for five bits on is 1F (11111 binary). The math looks like:

```
11010001000001101111101001110
                          11111
00000000000000000000000001110
```

Remember (1 & 1) = 1 and (1 & 0) = 0. So we have 0x1110 or 14 seconds. It was only necessary to mask to get the seconds. To get minutes, we'll have to shift and then mask.

For minutes, we have to shift over 6 bits to get past the seconds. D106F4E right shifted five times becomes:

```
00000110100010000001101111010
```

So now we mask another six bits:

```
00000110100010000001101111010
                        111111
00000000000000000000000111010
```

111010 is binary for 58. The rest of the conversion is left as an exercise for the curious. And no, I have not found out why the hours are off by 1. The following program was compiled with **Mark Williams C** (works with **Megamax** but watch out for printf(...lx...)).

```
/*        exdate.c          A  simple  example  of
converting   the   systems
internal time into human format*/

#include        "osbind.h"

main(){     unsigned long   time;
            int  secs, mins, hrs, days, mons, yrs;

/*   Grab system time                          */
    time = Gettime();

/*   Chew the bits                             */
    secs = ( time & 0x001f) << 1;
/*   bits   0:4                                */
    mins = ( time >> 5) & 0x3F;
/*   bits   5:10                               */
    hrs  = (( time >> 11) & 0x1F) + 1;
/*   bits  11:15                               */
    days = ( time >> 16) & 0x1F;
/*   bits  16:20                               */
    mons = ( time >> 21) & 0x0F;
/*   bits  21:24                               */
    yrs  = 1980 + ((time >> 25) & 0x7F);
/*   bits  25:31                               */

    printf("ST time %lx is %d/%d/%d at %d:%2d\n",
            time, mons, days, yrs, hrs, mins);
    Cconin();}
```

Mark Williams C

Mark Williams has a C for the ST on the market. Many of you porting IBM code will be very happy with the latest ST C on the market. For UNIX-philes, there is a C-shell like interface with most of the familiar tools.

Good news - The documentation is filled with interesting examples and explanations. The compiler is faster than Alcyon (not by much) and is a K&R with extensions (let's hear it for enums), and, so far, the least expensive of the serious C compilers on the market (approx. $140). There is an archived copy of the Microemacs source code in the distribution. A program is included to convert Alcyon object files to MW objects.

Bad news - This package is designed for either hard disk or two double sided disks. All the nice windowing calls are undocumented but MW recommends the Abacus GEM book. Almost as slow as the Alcyon compiler but it does compile and link when you 'cc'. The resulting .PRGS are larger than MegaMax's .PRGs.

While not a slick as Megamax, Mark Williams is a better choice for C programmers who cut their teeth on IBM PCS or UNIX systems. A good guess is that many of the **Mark Williams Cs** will be sold to those porting programs from IBMS.

Note: If you buy **MW C** and have one single sided drive AND one double sided drive, please do the following, and DO NOT run the install program:

1. Backup the disks.
2. Copy the .PRG files from the second disk to a single sided disk.
3. Copy the .A files (except for MicroEmacs archive) to the single sided disk.

Now you can run and be almost as happy as someone with two double sided disks.

NEXT MONTH

The editor sent me a note to say that he had a copy of **Lattice C** for me to evaluate. What I plan to do is to have a comparison of **Alcyon, Lattice,** Mark Williams, and **Megamax** a la Dr. Dobbs Journal.

P.S. - The USENIX news billboard has an Atari St section that gets a surprising amount of good software donated to it. If you can get to a system with a news feed please check USENIX out.

The Joys of Modula-2/ST

by Sol Guber

Just like a bad penny, I am back again. If you remember my last column, I showed you how to put a window on the screen. In this column, I will use that program as a module that will be IMPORTed. We will then manipulate that window on the screen. (For those who suggested opening it for a breath of fresh air away from my columns, I will not make any further comments). However, what we will do is move the window around and even make the window larger and smaller. These are operations that you have been doing with windows since you got your Atari ST but now they will be under your control.

This column will discuss EVENTS. The reason that events is capitalized, is because of the importance of events to the whole GEM scheme. GEM was built up around events. In the old days, pre Macintosh, a computer would only respond to a command that it was expecting. For example, if a letter choice was to be entered into the system, you could move the Joystick around all you wanted, and the computer would not recognize that movement. Things have changed however. With the ATARI ST, if the system is expecting a letter and you move the mouse to the top of a screen and press a button, there is a good likelyhood that a menu choice will drop down and will take over the screen. You can do several things with the system while it is waiting for the number. Events manipulation is what makes the ST a much more user friendly system than previous computers.

Events on the ATARI ST are divided into five groups: keyboard events, mouse button events, mouse pointer events, messages, and timer events. Keyboard events occur whenever a key is pressed. The mouse button event monitors which of the buttons has been pressed and how many times each have been pressed. The mouse pointer watches the changes in the mouse's position on the screen. The message event is used to show that the user has made a request of the system like moving the window or changing its size. These messages will be the major event that we will consider in this article. The timer events use one of the many timers in the system.

The procedure to use for monitoring events is called Eventmultiple and contains the following parameters. A usage is shown in Figure 1. The names for the parameters are my own. The parameters, in order, are event types, click number, mouse button type, mouse button number, rec1 flag, rec1 x, rec1 y, rec1 w, rec1 h, rec2 flag, rec2 x, rec2 y, rec2 h, rec2 w, messages, timer lo, timer hi, mouse x, mouse y, mouse button state, keyboard state, key pressed, and mouse button number. This is quite a mouthful and EventMultiple is easily one of the most complicated commands available. This is the command to check on all five types of events, and each of the event types can be monitored individually. The individual functions are Eventkeyboard, for key presses, Eventbutton, for monitoring the mouse buttons, Eventmouse for watching the mouse, Eventmessage, for the predefined message, and Eventtimer for the system clock. In actuality, there is just one procedure and all of the others call EventMultiple.

Now how does EventMultiple work? The first variable that needs to be entered is event types. There is a numerical code for each of the five

```
MODULE Happening;

FROM Window IMPORT InitWindow, CloseWindow,handle;
FROM AESEvents IMPORT EventMessage;
FROM AESWindows IMPORT WindowUpdate,WindowSet,WindowCalc,WindowGet;
FROM GEMAESbase IMPORT WindowRedraw, WindowTopped, WindowClosed,
     WindowFulled, WindowSized, WindowMoved,WindowNewTop,
     WindowHorizSlided;
FROM AESGraphics IMPORT GrafMouse;
FROM GEMVDIbase IMPORT PxyArrayType;
FROM SYSTEM IMPORT ADR;

VAR
    xdesk,ydesk,wdesk,hdesk : INTEGER;

PROCEDURE BigEvent();

VAR
    Msgbuff : ARRAY [0..8] OF INTEGER;
    mx,my,dummy : INTEGER;
    Fulled : BOOLEAN;
    xwork,ywork,wwork,hwork: INTEGER;
    xold,yold,wold,hold : INTEGER;

BEGIN
  LOOP
  EventMessage(ADR(Msgbuff));
    CASE (Msgbuff[0]) OF
      WindowRedraw : |
      WindowNewTop : |
      WindowTopped :  WindowSet(handle,10,0,0,0,0); |
      WindowHorizSlided :
        WindowSet(handle,8,Msgbuff[4],Msgbuff[5],Msgbuff[6],
              Msgbuff[7]); |
      WindowSized :
        WindowUpdate(1);
        WindowSet(handle,5,Msgbuff[4],Msgbuff[5],Msgbuff[6],
              Msgbuff[7]);
        WindowGet(handle,4,xwork,ywork,wwork,hwork);
        WindowUpdate(0);|
      WindowMoved :
        WindowUpdate(1);
        IF (Msgbuff[6] < 60) THEN Msgbuff[6] := 60; END;
        IF (Msgbuff[7] < 80) THEN Msgbuff[7] := 80; END;
        WindowSet(handle,5,Msgbuff[4],Msgbuff[5],Msgbuff[6],
              Msgbuff[7]);
        WindowGet(handle,4,xwork,ywork,wwork,hwork);
        WindowUpdate(0);|
     WindowFulled : |
     WindowClosed :   CloseWindow; RETURN; |
     ELSE;
     END;
  END;
    CloseWindow();
END BigEvent;

BEGIN
   GrafMouse(3,NIL);
   WindowGet(0,4,xdesk,ydesk,wdesk,hdesk);
   BigEvent();

END Happening.
```

types of events. The codes in hexadeciamal are: Keyboardevent = 01H, Buttonevent = 02H, Mouse1Event = 04H, Mouse2Event = 08H, Messageevent = 10H, and Timerevent = 20H. If you want multiple events to be watched for, you just add up the sum of the constants. If you want just the button and the messages to be monitored, event types equals 12H. This allows for the selective monitoring of the system. EventMultiple returns a number showing which of the events triggered the system. If events type was set equal to 12H and only a message event occurred then the value returned would be 10H. The returned value is always a subset of the possible events.

The next three variables have to do with the mouse and the buttons on the mouse. Click number is the number of times the mouse button has been pressed. Mouse button type tells which button is being monitored with the left button being 01H, second button from the left being 02H, and so forth. The Mouse button state shows whether the button is up or down with 1 being up and 0 being down.

The next five parameters in EventMultiple are used to determine changes in mouse position. A rectangle on the screen is defined and the system watches whether the mouse pointer moves enters or exits this rectangle. A one is used to signify enter and a zero is used for an event occuring when the mouse exits. The rectangle is defined by the next four parameters. The next five parameters are used to define a second rectangle on the screen which can also be monitored.

The next parameter in EventMultiple is the address where 16 messages from the system can be placed. Figure 5 shows a list of values used for the predefined messages that can be passed back to the program in Msgbuff[0] by the AES system. The rest of the Msgbuff will contain values pertinent to some of the messages. The only messages that we will be concerned about in this program is the messages about what is being down to the window, especially sizing it, moving it, fulling it, and closing it. The menu selected message will be discussed in the next column.

The next two parameters in EventMultiple are the value in milliseconds for the timer before it times out and signals an event. The parameters are in standard format, low byte, high byte. The final six parameters return values from the event caller. Mouse x and Mouse y are the actual positions of the mouse when the event occured. The mouse button state, when the event occured is returned. The keyboard state tells whether shift, control, or the alternate keys are pressed. The key press is used for the actual value of the key that was pressed. Finally Mouse button counts how many times the

Mach 2 for the Atari ST

Mach2: multi-tasking Forth-83 development system

With everything you need to develop stand-alone appplications, including: integrated GEM editor, full GEM and TOS support, Motorola assembler, debugger, demos, and our 300 pg. manual.

Mach 2 is interactive, so it allows you to experiment with the ST without going thru the compile-link-execute cycle. But when you do load in programs, look how we stack up:

Fast Sieve	Compile	Link	Execute
Mach 2 Forth	0:00.2	0:00.0	3.92
Megamax C	0:50	0:24	2.70
Mach 2*	0:00.2	0:00.0	1.41
Megamax*	0:50	0:24	1.41

*Using built-in Motorola assembler

(That's three times the execution speed of other Forth's) Note the turnaround time. It simply takes less time to develop your programs or finished products with Mach 2.

Palo Alto Shipping
PO Box 7430
Menlo Park, CA 94026
800/44-FORTH (Sales)
415/854-2749 (Dev. Support)
415/854-7994

*Price will be $99.95 as of Nov 1, 1986

all for only
$59 95*
plus $5 S/H
CA Res add 6.5%
VISA/MC COD

Original Macintosh version $99.95
Amiga version $99.95
EPROM systems available, too

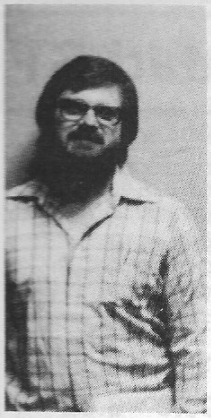
ST Applications
THE ATARI 520/1040ST JOURNAL

On Disk

For Vol. 1 (1985- 4 issues) of ST Applications the Support Disks contained both the text files for the articles as well as the programs we published. Public domain software and text files were added to fill up each monthly disk.

For Vol. 2 (1986- 9 issues to date) of ST Applications the Support Disks contain only the programs and support files as well as a good helping of Public Domain software each month. Text files for the magazine articles are not included on the monthly disk.

All 13 disks are available for $10 each ($12 with copy of ST Applications). Please make checks payable to ST Applications, P.O. Box 980, Forestville, CA 95436



Bill Petry

ST Applications
THE ATARI 520/1040ST JOURNAL

P.O. Box 980
Forestville, CA 95436
(707) 869-2325

Statement of Ownership,
Management and Circulation

Title	ST Applications
Frequency of Issue	Monthly
# Issues Annually	12
Annual Subscription Price	$40
Mailing Address	10760 Hwy 116 Forestville, CA 95436
Publisher	William Petry
Owner	William Petry
Total # of Copies Printed (12 month average)	3532

Circulation (average per issue for preceeding 12 months)

Dealers and Vendors	1712
Mail Subscription	224
Total Paid Subscription	1936
Free Distribution	208
Total Distribution	2144
Copies Not Distributed	1358
Return from News Agents	40
Total	3532

mouse button was actually pressed.

Now that we have the heart of the program, the event manager the rest of the program becomes easily explained in several sentences. First, draw a window with the proper parts (last month's column). Check to see if an event occured. Determine which event occured. Perform the proper action. Repeat as needed. There is nothing to it. It is just the details right! That is just what the program in Figure 2 does.

As with all Modula-2 programs, the first operation is to IMPORT the proper procedures for other areas. The program from last month must be redone so that there is a Definition Module containing the procedures Initwindow, Closewindow, and the variable handle, so that we know which window we are working on.

A new procedure is used in this program called Windowupdate. It is used to signal GEM that a job with the window is being performed and that the user has taken over the screen. It has four possible parameter numbers. One signals that an update is being done and no other function with the window will be allowed like no menu will be dropped. Zero signals that the update is finished and other operations can now be performed. The other values, 2 and 3, are seldom used.

The major part of the program is the Procedure Bigevent. Since the program is to be kept small, only a selected few messages from the window will be examined. I will not check for mouse or the keyboard events. Thus only the EventMessage procedure is needed to be called and it is much simpler than EventMultiple.

Not all of the possible messages that can be sent to Msgbuffer will be tested. I have listed all the possible ones in Figure 5, but many of the possible ones will be neglected. I will only write the code for four of the possible messages, the movement of the horizontal slide, the movement of the window, the change in size for the window, and closing the window.

An endless LOOP is used to keep checking for the events until the CloseWindow message is returned and the program stops. The Modula-2 CASE structure is used to check for each individual possibility. The form for the CASE is very similar to that of PASCAL except that the ELSE is used. The CASE form can be seen in the program. The CASE form is used rather than using a number of IF THEN ELSIF since it makes for clearer programs. The first message checked is that for a change in the horizontal slider. If that is the message, then the Windowset command is used to change the position of the horizontal slider.

The next message that is checked is that for a change in size of the window. If this message is

can be seen in the program. The CASE form is used rather than using a number of IF THEN ELSIF since it makes for clearer programs. The first message checked is that for a change in the horizontal slider. If that is the message, then the Windowset command is used to change the position of the horizontal slider.

The next message that is checked is that for a change in size of the window. If this message is true, then the system is notified that a window update is in progress. The WindowSet command is used to make the current variables of the size the same as those returned from Msgbuff. These are then make part of the current set and then the window update is set to zero to show that it is over. If you use this procedure, you will notice something strange on the screen. The inside portion of the window is not changed. It is the same as it was before the the window was changed in size. The programmer herself must make user that the proper information is put back into the window when the size has been changed.

If the window was moved, check to make sure that the sides and height are still the proper size. Then perform a Window set with the proper variables for the current values. Finally perform a WindowSet with the proper variables for working variables.

If the Windowclosed message was sent then the window is closed and the program is over. The CloseWindow procedure is the one used in last month's column.

The initialization statements for this program are also simple. First use Grafmouse to change the shape of the mouse to a pointing finger. Determine the size of the screen. Finally call BigEvent. This is an endless loop and does not complete until the close window command is clicked.

I hope that you have enjoyed moving your window around on the screen. It is not very difficult. Next month, I will also the the event monitor to use the menu and make your own menu's.

More FORTH from page 31

promise of suiting somebody; whether it will agree with you all depends. Future topics I am considering include a description of the BIOS, XBIOS and sound routines. Anyone wishing to suggest further FORTH curiosities should address all correspondence to Albert Lew, 680 Blair Ave, Piedmont, CA, 94611. Until, then happy VDIing!

Atari ST heats up with "Strip Poker"... and more.

Examine Artworx' expanding list of ST software:

STRIP POKER is *the* classic computer program. Play against Suzi and Melissa; the more you win, the more they take off! **Only $39.95**

THAI BOXING brings oriental kick boxing to the ST with stunning 3-D graphics and fast action. **Only $19.95**

HOLE IN ONE GOLF plays like the real thing and includes an easy-to-use Editor for designing your own golf course. **Only $29.95**

BRIDGE 4.0 has full graphic display of all hands, allows the user to be dealt good cards all the time and is completely mouse-driven. **Only $29.95**

COMPUBRIDGE is Artworx' bridge tutorial, utilizing 10 chapters covering all aspects of the game. **Only $29.95**

PEGGAMMON is a new and innovative approach to backgammon. Play against the computer or with a friend in the two-player mode. **Only $17.95**

MAILLIST handles medium to large sized lists. It zip code sorts, alphabetizes, selects by keyword, prints labels, and more. **Only $17.95**

BAKER STREET DETECTIVE brings you back to 1893 London with mysteries to solve just like Sherlock Holmes. **Only $17.95**

Artworx®

Artworx Software Co., Inc., 1844 Penfield Road, Penfield, N.Y. 14526 (716) 385-6120 · (800) 828-6573

Forth Programmers

Are you tired of buying the latest drawing program only to find out that it falls short of your expectations?

Want to write your own CAD program, but don't want to spend a month writting all of those basic but necessary routines? Then get a copy of

FORTHCAD™

Included on the disk is a click-on application as well as a complete and fully commented source code listing (over 350 screens), written in 4XFORTH* for all of the usual drawing tools, as well as 1 and 2 point perspective, full screen crosshairs and rubber arcs between points.

For 520ST 2 single-sided disks $35.00.
For 1040ST 1 double sided disk $33.00.
Future updates $12.00 to registered owners.
Send checks or money order to:

FORTHWARE

5360 126th St., Hawthorne, CA 90250
For technical information or to order by phone: call (213) 643-7106

*Trademark of The Dragon Group Inc.

LOGO LOG

by Leonard Kaplan

Wanna Buy Some Properties, Cheap? Or, A Unique LOGO Feature

A quick note: this column will make much more sense to you if you do the exercises as you read the column. Usually any given example will depend upon something done in a previous example. I'll try to avoid making examples carry from month to month, though, so as not to leave new readers out in the cold. This month we're going to look at an unusual feature of ST LOGO, one that the "usual" home computer languages don't have. Over the next few months, we'll use this feature as part of a basic inventory system. The system I'll implement here will be for a VERY small hobby shop, but the basic concepts could certainly be applied to other applications. So feel free to use my code to help organize your record or tape collection, or your disk library (not that THAT needs to be organized, right?). Let me know what you come up with, maybe other readers would be interested.

Objects in LOGO can be assigned properties. A property is exactly what it sounds like - a characteristic of the object. ST LOGO allows the programmer to go far beyond the usual high-level language "properties" like "it's a table" (the object was DIMENSIONed), or "it must be a day of the week" (assigning an enumerated value in PASCAL). For example, type in the following (you ALWAYS have your system up while reading ST APPLICATIONS, don't you?) - the computer's responses are underlined:

```
?ERALL
?MAKE "MYLIST [LUCY DESI FRED ETHEL]
?PPROP ITEM 1 :MYLIST "SEX "FEMALE
?PPROP ITEM 1 :MYLIST "LASTNAME "ARNAZ
?PPROP ITEM 2 :MYLIST "SEX "MALE
?PPROP ITEM 2 :MYLIST "LASTNAME "ARNAZ
?PPROP ITEM 3 :MYLIST "SEX "MALE
?PPROP ITEM 3 :MYLIST "LASTNAME "MERTZ
?PPROP ITEM 4 :MYLIST "SEX "FEMALE
?PPROP ITEM 4 :MYLIST "LASTNAME "MERTZ
?
```

Now, type in the procedure in LISTING 1. You don't need to type in the comments (they follow the ";"), they're just there to help you understand the code. Then, type:

```
?SHOWFEMALES :MYLIST
LUCY ARNAZ
ETHEL MERTZ
?
```

Interesting, eh? You're probably thinking "Gee, this could be useful! Just last week, the owner of a local computer store asked me to design an inventory system for him!" Or, maybe you're not. Anyway, here are the important built-in procedures related to property management:

1) GPROP object-name property-name

The GPROP procedure is used to output the value of a given object's property. The object can be a single object, as in the code for SHOWFEMALES, or it can be a list. If the object doesn't have the requested property, an empty list "[]" is returned.

2) PPROP object-name property-name property-value

The PPROP procedure is used to assign a property, and a value for the property, to an object. Again, this object can be a single object, as in the example above, or it can be a list. In the latter case, the list has the property, but the individual object does not:

```
?PPROP "MYLIST "LISTTYPE "TVPEOPLE
?GPROP "MYLIST "LISTTYPE
TVPEOPLE
?GPROP ITEM 1 :MYLIST "LISTTYPE
[]
?
```

3) PLIST object-name

The PLIST procedure outputs the properties of an object:

```
?PLIST "MYLIST
[LISTTYPE TVPEOPLE .APV [LUCY DESI FRED
ETHEL]]
?
```

4) REMPROP object-name property-name

The REMPROP procedure removes the named property from an object:

```
?REMPROP "MYLIST "LISTTYPE
?GPROP "MYLIST "LISTTYPE
[]
?
```

There are two other property-related procedures in ST LOGO, but first, a brief explanation.

Procedures may be put together in groups called PACKAGEs. For example, to put the procedure SHOWFEMALES and another one called SHOWMALES (I'll leave that as an exercise for you to write, but you'll need it to continue typing in the items in this article!) into a package called PROPERTYDEMOS, you type:

```
?PACKAGE "PROPERTYDEMOS [SHOWFEMALES
SHOWMALES]
?
```

To put all of the items that aren't packaged already into a package called OTHER, type:

```
?PKGALL "OTHER
?
```

There are other package-related commands, but I'll save them for a future column. Continuing with the property procedures:

5) GLIST property-name package-name/package-list

The GLIST procedure returns a list of the objects in the workspace or specified package(s) that have the requested property. For example:

```
?GLIST "SEX "OTHER
[ETHEL LUCY DESI FRED]
?GLIST "FOO "OTHER
[]
?PPROP ITEM 1 :MYLIST "CRAZY "YES
?GLIST "CRAZY "OTHER
[LUCY]
?
```

Note that GLIST doesn't give the value of the specified property, it just notes which objects have it.

6) PPS property-name package-name/package-list

The PPS procedure displays the non-system properties of all objects in the workspace or the specified package(s). For example:

```
?PPS "OTHER
ETHEL'S LASTNAME IS MERTZ
ETHEL'S SEX is FEMALE
LUCY'S CRAZY IS YES
LUCY'S LASTNAME IS ARNAZ
LUCY'S SEX is FEMALE
DESI'S LASTNAME IS ARNAZ
DESI'S SEX is MALE
FRED'S LASTNAME IS MERTZ
FRED'S SEX IS MALE
?
```

```
TO SHOWFEMALES :PERSONLIST ;PERSONLIST IS LIST OF FIRST NAMES
;
LOCAL "MAXDO ;ONLY AVAILABLE IN SHOWFEMALES
LOCAL "INDEX
LOCAL "CURRITEM
;
MAKE "MAXDO COUNT :PERSONLIST ;HOW MANY PEOPLE TO CHECK
MAKE "INDEX 1 ;OBJECT LIST INDEX
;
LABEL "SHOWFEMLOOP
;
MAKE "CURRITEM ITEM :INDEX :PERSONLIST ;OBJECT TO EXAMINE PROPERTIES OF
IF (GPROP :CURRITEM "SEX <> "FEMALE) [GO "NOTFEM] ;BYPASS IF NOT FEMALE
TYPE :CURRITEM ;ELSE TYPE WOMAN'S FIRST NAME,
TYPE CHAR 32 ;A SPACE,
TYPE GPROP :CURRITEM "LASTNAME ;AND HER LAST NAME.
TYPE CHAR 10 ;GO TO NEXT TEXT LINE
;
LABEL "NOTFEM
;
MAKE "INDEX :INDEX + 1 ;BUMP POINTER,
IF (:INDEX <= :MAXDO) [GO "SHOWFEMLOOP] ;AND DO AGAIN IF NEED TO.
END
```

There's definitely potential in ST LOGO for more than turtle graphics!

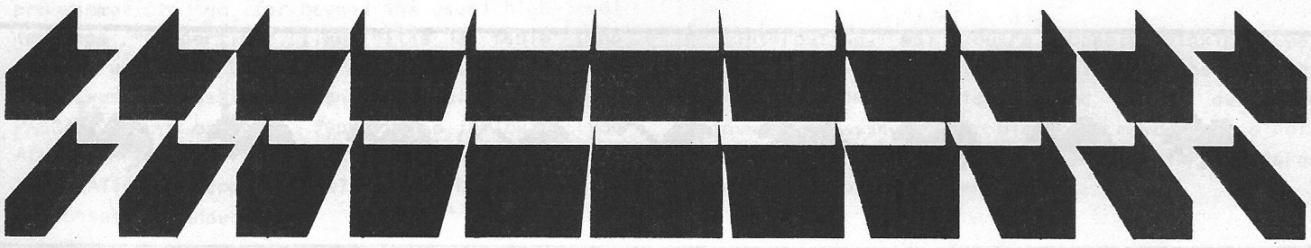
A LOOK TO THE FUTURE

Here is my idea of what the (upcoming) tiny inventory system should be able to do:

1) Create a stock file;
2) Add new items into the file;
3) Add and remove stock;
4) Allow "sale" items;
5) Let a customer buy items by using a shopping list of some kind (no bad joke intended there); and
6) Print an inventory list.

Keep in mind that this is really just a first draft. I am considering expanding the system to allow "multiple-warehousing". This means that the store could have items stored in multiple locations (files). For example, there could be 20 paintbrushes in WAREHOUSE1 and 15 in WAREHOUSE2. Recently, a company here in upstate New York spent large amounts of cash adding this feature to an existing inventory system - obviously, it may not be a trivial thing to patch in. If designed in (or hooks put in) right from the start, though, it is a nice feature - have you ever been in a "catalog showroom" and had the cashier tell you that the item is available at another store in the chain - without calling the other store?

A call to my readers for some information. Is there any way to expand LOGO to have more than 256KB of workspace? If you type in the NODES command, it won't come back with more than about 63000 nodes - about 252KB. This is on a 1040ST with just the control panel installed. Would more workspace be useful? Or more stack space (this is another unknown)? You tell me.



The Beginner's Corner

by Tom Hawkins

DESKTOP Setup:

Question: Why be concerned about DESKTOP setup since everything will function pretty well no matter how you set it up?

Answer: Some of us want to maximize productive time and don't like to "waste" time waiting for the computer to "catch up".

Say, what? That's right! If you have to spend time moving windows around and waiting for programs to load from disk (more than once) then your computer is running you instead of vice versa. A striking example of this is Print Master (Unison World), which can be excrutiatingly slow when floppy run, or quick-paced when Ramdisk run.

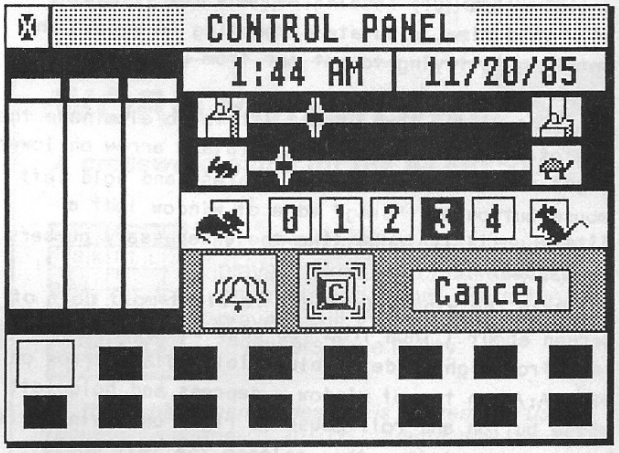
Everybody should have a certain setup they use. Some may have more than one, depending on what type of projects they're working. My standard setup is designed to be aesthetically pleasing and fast, particularly for the type of work I do most (wordprocessing and printing). If you'd like a format that works pretty well, install the following (else skip to something more stimulating):

Before you start, set up your disk with an Auto folder containing JDISK.PRG and RDCPY.PRG (requires J-Disk from Cal Com Inc.), a folder named RDCPY (put STWriter with XYZZX.DAT in there for now), and CONTROL.ACC. If you've got other stuff on the disk it may affect the outcome.

NOTE: Your inputs are underlined
<click> means click the left mouse button
<return> means press return key

"My" setup:

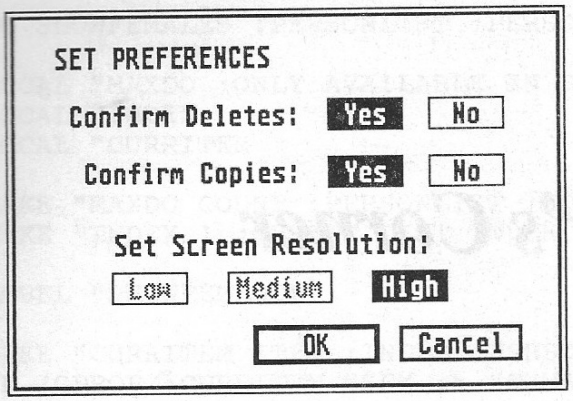
1. The tools - J-Disk (Cal Com Inc.), ST Writer (ATARI), GEM Desktop (part of the ST operating system).



2. The GEM Desktop settings:

			CHANGE	
		Setting	From	To
Desk: Cntrl Panel:	a.	White	777	642
	b.	Black	000	211
	c.	Green	060	007
	d.	Red	700	070
	e.	Rabbit-turtle slider, move to far left.		
	f.	Upper left corner of panel <click>		

File: No change
View: Show as text <click>



Options: Set Preferences:
a. Confirm copies NO <click>
b. OK <click>
Save desk top <click>
Set Preferences:
a. Set Screen Resolution MEDIUM
b. OK <click>

3. The DESKTOP:

a. Put mouse arrow on "A" Floppy Disk ICON - <click click> (if nothing happens try it again, only this time, simulate attempting to squash an ant that is trying to get out from under your fingertip).
b. Size window down just a little to eliminate the row of 2 numbers on far right (place arrow on lower right corner box of window, depress and hold left mouse button and "drag" edge of window left a little until it looks like the unnecessary numbers "disappear").
c. Drag the window (intact) to right-most part of screen about 1 inch from top edge and about 1/8th inch from right side of blue field (place arrow on the "A:/" on top of window - depress and hold left mouse button and roll mouse to right until window is where you want it - then release the left mouse button).
d. Trash ICON - drag to a point about 1 1/2 inch left of the "A:/" window you just created and about 3/4 of the way down from top of your screen.
e. Eliminate "B" Floppy Disk ICON - put arrow on B Floppy Disk ICON <click>. Move mouse arrow to OPTIONS then Install Disk Drive...<click> Remove <click>.
f. Install "D" Ramdisk - put arrow on "A" Floppy Disk ICON <click>. Go to OPTIONS and Install Disk Drive...<click>.
Now:
(1) Hit <esc> key and type a capital "D"
NOTE: If you hit <return>, start again at f
(2) DON'T hit <return> - this is for you speed readers
(3) Press once on "down" arrow (it's under the "up" arrow which is between the "Insert" and "clr home" keys, right under the "Help" and..... oh, good - you found it) then press <esc>
(4) Type in "Ramdisk"
(5) See note in (1) above
(6) Place mouse arrow on Install <click>
(7) Drag "D" Ramdisk ICON to immediate right of "A" Floppy Disk ICON
(8) Double-click on "D" Icon
(9) Drag the window for "D:/" over next to "A" window so the tops are even and about 1/8 inch separates the two. This should about cover the trash ICON, but not quite ("TRASH" should still show)
(10) Size the "D:/" window up so that about 1/2 of trash can shows.
(11) Place arrow in "A:/" window (anywhere) and <click>
(12) Place arrow on OPTIONS and Save Desktop <click>

And there you have it. It takes a little time and thought to set up the first time, but, once you've done it, the computer does all the work from then on. It takes about 50 seconds from "power-on" to system "ready".

RAMDISKS:

There are a couple of commercial versions I use, J-Disk (Cal Com Inc.) and M-Disk (MichTron). You can use any that works well for you.

J-Disk is neat because on boot up it lets you set date and time and loads selected programs. Drawbacks to J-Disk is that the Ramdisk it creates always seems to be named "D" and when sized too small for the RDcopy contents, RDcopy will copy all the files (in name only) to the ramdisk truncating the files to the total ramdisk size making them useless.

M-Disk is neat because it lets you add as many Ramdisks as you have RAM for and designates a new letter for each new Ramdisk you install (i.e. E, F, G, etc). I got up to "I" and quit (100K at a whack). Also, M-Disk can be installed anytime - so, if you're chugging along just doing wordprocessing or something equally worthwhile and decide you want to play a game but aren't sure which one; create a 400K Ramdisk and load in Star Raiders and Joust or Time Bandits (DEMO), then you can call a game from Ramdisk and play until your kids start coming down the hall, then you quickly switch back to the mundane task you were previously

engaged in (don't forget to save your score) and your "image" as hard working slave to computer tasks is preserved.

How do I use Ramdisks? Well, for starters I use J-Disk with 100K on initial boot-up to load STWriter, Procopy (PROCO Products), M-Disk, and assorted StWriter files for letter writing and business use.

If I decide to get heavier into projects like printing with Print Master (Unison World) or GAMES(?), I'll add 400K with M-Disk and load everything I can into "C" Ramdisk. That gives me great flexibility to switch quickly back and forth as necessary.

A very practical application is to put frequently used terms, addresses, etc. into the same Ramdisk that STWriter is in, then you can use "MERGE FILE" to input them as necessary. For example, I have most software titles along with manufacturer's name and address in a file. When I MERGE and call up PM.SHT I get "Print Master (Unison World)". PM.LNG gets me "Print Master (Unison World, 2150 Shattuck Ave. Berkeley, CA 94704)", and PM.ADD produces:

Unison World
2150 Shattuck Ave.
Berkeley, CA 94704

Saves in way of "look up" and "type in" time. You do need to take the time for initial set-up - and correct errors whenever you find them.

Closing notes:

In using ST Writer, you can access all files. The way to do it is to follow the "path". For example, if I'm using "C" Ramdisk and decide I need a file from "A" disk, I need to know what file it's in. If it is in A:AUTO:STAP:CREDITS, which is the CREDITS folder in the STAP folder in the AUTO folder on "A" drive, then when ST Writer asks what file to print to screen for index, type A:\AUTO\STAP\CREDITS. This will show you all files in the folder CREDITS. When you get the name of the file you need, press "L" for load, then A:\AUTO\STAP\CREDITS\file name. Once you've set up the path you can access the folder whenever you want - until you change it. "C:" plus file name is sooo much easier, that's why I load what I plan to use into the same ramdisk folder as ST Writer.

PROCOPY
ST BACKUP UTILITY
$34.95
★ DUPLICATE VIRTUALLY ANY DISK
★ PROCOPY WORKS WITH ALL ST S/S & D/S FLOPPY DISK DRIVES
★ PROTECTS AGAINST ACCIDENTAL LOSS OF EXPENSIVE SOFTWARE
★ NOT COPY PROTECTED
★ UPDATING POLICY: $10 W/RETURN
SOFTWARE IS EASILY DAMAGED BY THE ELEMENTS AND HANDLING. YOU WANT TO BACK UP YOUR INVESTMENT BUT COPY PROTECTION LOCKS YOU OUT!
PROCOPY IS THE KEY!
★ ★ ★ ★ ★
SEND CHECK OR MONEY ORDER PLUS SHIPPING
$2.00 NORTH AMERICA, OVERSEAS $4.50
DEALER INQUIRIES WELCOME
PROCO PRODUCTS
P.O. BOX 665, CHEPACHET, R.I., 02814, U.S.A.
TO ORDER CALL (800) 843-1223 OR (401) 568-8459

WORD FOR WORD™ ST
A crossword game for the ATARI ST!

It's a challenging game in which players take turns creating words on a playing board. Scrabble® players will love it! Here's what reviewers have to say:

"...the whole game design is extremely user-friendly...a winner." ANALOG COMPUTING, June 1986

"...It's easy to use the mouse to design and save your own board layout...makes the game even more fun." ANTIC, April 1986

"...I am very impressed with Word for Word...full utilization of GEM...solid performance...a joy to play...attention to detail...excellent product." ST APPLICATIONS, Jan.-Feb. 1986

To Order

Contact your Atari ST dealer, or send $39.95 plus $3.50 for shipping and handling. ($43.45) California residents add $2.40 sales tax. ($45.85)

MasterCard or Visa accepted

Bay View Software
177 Webster St., Suite A-295
Monterey, CA 93940
(408) 373-4011

Works with color (medium resolution) or monochrome monitor.
Scrabble is a registered trademark of Selchow & Righter Co.

Review

A FinalWord in the Hand

by Donavan Vicha

Certain End Users have aired their fears that the ST is not being used for business purposes, but I know that my wife and I have found the ST an invaluable tool as a word processor. When we bought our computer, we also bought The FinalWord because it seemed the only full-featured writing software available at the time. It remains so eight months after we took the plunge into the world of Atari.

I'm not going to review Mark of the Unicorn's powerful word processor as much as make recommendations about its usefulness and how to get the most out of it. I have spent more than 250 hours working with this program, and I wish I could say that all of my time was spent processing words. I can't. But if what I've learned is worth sharing, then the time was well spent. I'll let you be the judge.

An Overview of The FinalWord

The FinalWord is for serious, heavy-duty writing. We're talking contracts, books, proposals, form letters, thesis papers, etc. It is priced and packaged for business use, both large and small. You can use it for simple letters and notes to friends, but it requires a certain amount of time setting up and shutting down. You certainly shouldn't buy it for such simple tasks.

The version I'll be discussing (V1.17) has been reviewed by experts at **ST Applications**, **Antic**, and **Analog**, and not altogether favorably. All reviews acknowledged, the myriad features of FinalWord are worth investigating. It is incredibly versatile with full formatting capabilities to produce any kind of document you could desire, except for double column printing. It does include indexing, contents numbering, footnoting, headers and footings, multiple printer drivers and driver configuration program, lists, endnotes, keyboard customization, and multiple-window editing. The keyboard commands are not always simple, but learning to use the editor is made easy by an excellent tutorial.

The reviewers' main criticisms of The FinalWord focus on the buffer management system and the Reference Manual's generic documentation. I agree they could be better. The sales copy reads: "You've got your hands on the only word processor ever written specifically for writers." That may well be true, but after many, many hours of experimentation and calls to MOTU's support staff, I can tell you that FinalWord wasn't written specifically for the ST. If you have a computer that is equipped with the friendliest graphics environment around, you would like to use that environment. The FinalWord doesn't. Only the four-page insert, which deals mainly with installing the programs, makes any mention of the ST. And that installation process bypasses the GEM data files.

I had to do a lot of experimenting to find out how to carry out what I think of as the housekeeping chores: setting up the program, saving and retrieving files, opening new files, printing--that sort of thing. The irony here is that housecleaning is carried out on the GEM desktop and the four-page ST insert gives no guidance in that direction. With the tutorial instructing you to type "fw" or "b:test.doc", you can get awfully confused.

I also agree with the critics that the buffer management system, the virtual memory/swap file set-up that makes the program crashproof, also causes some annoyances: a "Swapping" message and clearing of the screen if you aren't typing after ten seconds and limited memory for files in the default mode of the Recovery program, the program

that creates the Swap file where your text is stored in case of a crash. There is also the dual disk aspect of the program that almost makes having two SS drives or a DS drive a necessity.

Since I had no choice but to learn how to master the software in order to get a freelance job done, I have come to appreciate the strengths of The FinalWord and the fact that a word processor in the hand is worth more than those others still hiding in the bush. Here are some suggestions that have made using The FinalWord more tolerable for me. I can reservedly recommend FinalWord, high price and all, if you take these extra measures.

Ram-Charging The FinalWord

Using a ramdisk can really speed up any disk intensive program and The FinalWord certainly fits that category. You can overcome some of the annoyances that go with using a copy-protected program ("insert key disk") especially when using a single drive when two are really necessary. Install the C Ramdisk as whatever software you're using instructs, then put in your work disk, double click, and by holding the Shift key down, click on all the necessary files and copy them to the ramdisk.

The next step of installing applications (see page 48 of your ST Owner's Manual for full details) goes one file at a time for the .PRG files of Disk 1. You will install them as "TOS--takes parameters" files, which means they do not use the GEM features. Then insert Disk 2 in Drive A and copy the files not suffixed by .DOC (except for the one[s] you plan on using) and install the .PRG file same as the others.

Using the ramdisk, you only have to go through the installation process once, then save the desktop and remember to transfer that Deskinf to the #1 disk. From then on, everything is automatically installed as "TOS--takes parameters."

Now insert one of the original disks, open the window for Ramdisk C, double click FW.PRG, and you should be rolling. After the key disk copy protection sequence has been satisfied, you should remove that disk and put in the work disk. If you've been using FinalWord without installing the .PRG files (which is the four-page insert's SOP for single disk drive users) or keeping the GEM.ACC files, you are in for a treat. Watch that cursor just fly when you hold down the keys! No real need for using the two-key CR commands. The next thing you'll notice, or rather, not notice, is the "Swapping" line in the lower right corner of the window; it becomes much less of a bother than previously. And so on down the line, saving a file and printing files become smoother and faster.

Just remember that you will occasionally have to save and copy your document to a real disk, or it will disappear with the rest of your ramdisk when the computer is shut off or accidentally crashes. Transfer is quick and painless, too, so you can make as many back-ups as gives you peace of mind.

I confess that most of my experiments with this program occurred before I started using the ramdisk and that the crashproof capabilities saved a lot of text at times. Of course, I may not have had to do so much experimenting had I had clearer documentation and had the buffer management system been a bit friendlier. Now that I have greater control over all the power of The FinalWord, I think I know my priorities here.

Other Tips and Tricks to The FinalWord

To anyone interested in working on large projects, I have a few more suggestions that should make life with The FinalWord a little less adventurous and more productive.

EXPANDING THE SWAP FILE: The instructions for this manuever are covered on page 218 of the Reference Manual and work fine if you own an IBM. Since you don't, here's what MOTU staff told me. With the RECOVER.PRG installed as @b[TOS-takes parameters], double click on same. You'll get a window that asks for parameters.

Type in:

```
-create 96
```

This gives you a Swap file of about 100K. You can go up to 248K in increments of 8 pages if you like, but then you're really getting into massive files and a lot of searching around during editing. The default setting, however, is just a mite tight, so give the 96-pager a try. You can batch files if your document runs longer than the 50 pages this will allow.

Ian Chadwick, in his **Antic** review, complained that ST Writer has larger file capacity than The FinalWord and it's true, but who wants to scan through a 100-page document onscreen? Better to break things up into parts, which can be done by using the @@include command that allows merging of files. This is fairly well explained in the Reference Manual.

Another, more legitimate gripe that Chadwick raised is that whatever was onscreen the last time you used the program reappears when you load the FW.PRG. Unless you wish to continue working on it, you have to delete the buffer. If you had been working with more than one buffer, you have to delete them; they're all in there somewhere. If you don't, you could be facing a "Swap File Full" message when you least expect or want it. [Note: This can be eliminated by changing the default

setting in the CONFIG.PRG, as can the Swap timing mechanism. If you have two drives, do it and don't bother with a ramdisk. MOTU assured me the manual covers this adequately with proper study.]

Unless I am returning to work on a file or files, I usually don't copy the FW.SWP over to the ramdisk. It's a big file and takes a second or two to copy. It's faster just to run RECOVER.PRG and create a new Swap file as I just described. You start with a clean slate and save a millisecond along the way.

PRINTING INDIVIDUAL PAGES FROM A FILE: It's happened to me: I initiate printing and haven't got enough pages loaded in my printer. So, after loading more paper, I want to print the last five of twenty. The Reference Manual says this can be done (see page 321), but for the ST one tiny little command has been left out. When you tap F7 for Advanced Print Options, it tells you you type in "-page ##." Close, but no banana. You must type "-print -page ##." These small things shouldn't be worth a phone call to Cambridge, Mass. But there you go.

CONCLUSION: It would seem that setting up a ramdisk just adds to the ordeal of setting up The FinalWord, but the gains in speed of moving the cursor, ease of file management within and on the desktop, and speed and ease of saving and printing a file are worth it. Paying for a ramdisk rather than a second disk drive is certainly worth it.

By summer's end, we've been promised word processors that were designed specifically for the ST, and maybe, specifically for writers, too. It's a matter of taste as far as which one is for you, but it's also a matter of what you get used to using. I've put a lot of time into The FinalWord and now I feel it was worth it. I can wait for the other word processors to come out, the reviews to come in, the bugs to be shaken out, and maybe even a price war or two during the Christmas season. Needless to say, I'm writing this long-winded review and plenty of other long-winded documents with The FinalWord now that I've mastered its less-than friendly buffer management system. (I hope it's clear that actually writing and editing with The FinalWord is much easier to master and, in its way, fun.)

I just wonder why Mark of the Unicorn has let The FinalWord lay around without attempting to enhance it. No thesaurus or spelling checker? No attempts to follow the Desktop Publishing trend with mutiple fonts and graphics? The company seems to have squandered their lead over other software companies offering more state-of-the-art word processors for less money. Those who have paid the hefty price for The FinalWord deserve some effort toward improving the product. As things stand, it is a worthy piece of software, but the last word in word processing has yet to be written.

A Powerful Configuration Program and a Multitude of Printer Drivers

The FinalWord did not come with a printer driver for my SG-10 printer and I found both the Reference Manual for The FinalWord and the SG-10 very intimidating. The staff at MOTU was very helpful in this matter and sent a checklist of the modifications necessary to convert their built-in Epson driver to drive the SG-10. Most of the 28 items on the checklist didn't need to be changed and the operation was fairly easy.

The Configuration program for The FinalWord is the largest .PRG file on either disk. It allows you to customize just about any aspect of its various programs to whatever parameters you would like. It begins with a main menu of four options:

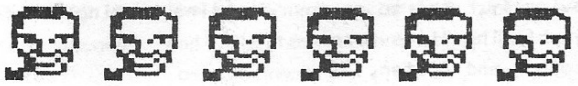
1) Select terminal type,
2) Select printer type,
3) Advanced configuration,
4) Exit.

There are nearly 50 printer drivers listed for The FinalWord and probably more available from MOTU. The Advanced Configuration menu provides a way to modify an existing driver to the specs of your own printer. The menu items are listed in the Reference Manual so that you can study their meanings and adapt them to what your printer manual specifies. The menu is fairly detailed in explaining what each option to be changed does.

There are other items on the Advanced Configuration menu as well:

1) Define terminal types,
2) Define printer types,
3) Define Input/Output ports,
4) Select default editor parameters,
5) Select default formatter parameters,
6) Define keyboard uses,
7) Return to main menu.

This is wonderful, albeit complicated, stuff here and probably from the items listed you can guess what can be done. For instance, if you are going to use The FinalWord for typing up contracts, the default formatting options can set things up so that you needn't use the @Style commands as heavily as would be needed in the default mode currently set. Ditto for the editing routines. [Note: There is an option for printing while editing that requires a BIOS call 15, LISTST, to work properly. If the ST does have this, MOTU has left out an important option and sales feature for The FinalWord.]



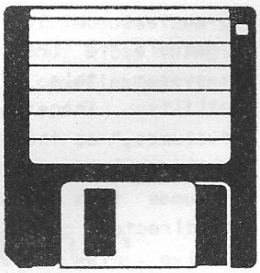
The Define keyboard option carries this menu of options:

1) Enable Function keys,
2) Bind one key to another,
3) Bind keys to actions,
4) Define ranges of keys,
5) Display key to action bindings,
6) Display action to key bindings,
7) Read key definitions from file,
8) Write key definitions to file, and
9) Return to submenu.

I am certain a true power user would find these options terrific, or at least understandable--me, it's still Greek, but I'm easily impressed by things I don't understand. The details of each of these menus are described in the Reference Manual in the Installation section.

Perhaps if there is enough interest in advanced uses of The FinalWord from **ST Applications** readers, I can be persuaded to continue experimenting with these features. Otherwise, I've found that I can process words to my heart's content with what I've already presented in this seemingly neverending article.

ST Applications
THE ATARI 520/1040ST JOURNAL



On Disk

For Vol. 1 (1985- 4 issues) of ST Applications the Support Disks contained both the text files for the articles as well as the programs we published. Public domain software and text files were added to fill up each monthly disk.

For Vol. 2 (1986- 9 issues to date) of ST Applications the Support Disks contain only the programs and support files as well as a good helping of Public Domain software each month. Text files for the magazine articles are not included on the monthly disk.

All 13 disks are available for $10 each ($12 with copy of ST Applications). Please make checks payable to ST Applications, P.O. Box 980, Forestville, CA 95436

It's Here!

NEW

BASIC COMPILER

only $69.95

For Atari 520/1040 ST

- So easy to use!
- A menu driven GEM application!
- Fully ST BASIC compatible!
- Produces small efficient code!
- Generates a stand alone program!
- Supports GEMSYS and VDISYS calls!
- Has hooks to BIOS!
- No line numbers needed
- Double-precision floating point numbers fully implemented

For the BEST professional BASIC Compiler on the market today send:

personal check (wait 10 days to ship), money order, cashiers check, VISA and M/C, C.O.D. (add $2.00). (California residents add applicable sales tax.)

— Dealers Welcome —

LDW **Logical Design Works, Inc.**
780 Montague Expwy., Suite 205
San Jose, California 95131
(408) 435-1445
Telex: 294526 LDW UR

ST-Copy

The Ultimate Backup Utility for the Atari "ST" Series Personal Computers

ST-Copy's multi-option duplicating and fast reproduction capabilities make it the ultimate back-up system for your computer. ST-Copy has been tested by the most challenging protection schemes on the market. We are proud that ST-Copy surpassed the highest standards set for software duplication. CAL COM is confident that you will be equally impressed with its performance.

ST-Copy is compatible with monochrome, RGB monitors, one or two disk drives, single or double sided.

Updates of ST-Copy will be available for $10.00

Order Now!...$34.95 (Dealer inquiries welcomed)

To Order Call or Write CAL COM INC.

EAST COAST (301) 681-9121
P.O. Box 2601
Silver Springs, MD 20902

WEST COAST (714) 523-5353
6820-A Orangethorpe Ave.
Buena Park, CA 90620

The Review Board

Currently FORTRAN compilers for the ST are available from three different sources: TDI, Pecan, and Prospero. Both the TDI and Pecan compilers are based on the USCD p-system; in fact, according to TDI they are the same compiler. The Pecan version is reviewed here. Neither it nor Prospero's FORTRAN is perfect, but both are useful products.

P-system FORTRAN

PECAN Software Systems, Inc.
1410 - 39th Street
Brooklyn, New York 11218

Pecan's FORTRAN is a product that has had a longer lifetime than the Atari ST. This is possible because of the UCSD p-system. The p-system is a virtual machine that can be implemented on many different computers. The p-system has been around since the late 1970's and was designed to support Pascal, in a multi-tasking environment. Most people who have used the p-system are either fanatical about it or don't like it at all. When using the p-system, be prepared to forget the desktop ever existed; you won't see it, icons, or menu bars anywhere in the p-system. In fact, there is a small program to remove the AES from memory if you still use a disk booted system.

Although this is a review of a FORTRAN compiler, I think it is necessary to review the p-system user interface, as well as the compiler itself. If you have used the p-system on other computers, it won't be necessary to learn anything new. The p-system looks the same on every system it runs on. Also be prepared for a little surprise. Since the p-system is so efficient with memory, it has plenty to spare, and formats the spare memory into a ramdisk. This also causes an irritating problem: the p-system is continually swapping segments of code in and out of the disk drives, slowing the system down. Personally, I would rather have a smaller or non-existant ramdisk and an increase in speed.

The p-system is a thoroughly integrated environment, along the same lines as the Turbo Pascal environment on the IBM PC. This means that it is possible to go from an error in a compilation to the proper line in the source code to correct it. Unlike Turbo Pascal, it is not possible to run p-system programs outside the p-system environment. In this way, it is similar to BASIC: there is a long bootup process before any program can be run.

Like some Forth interpreters, the p-system has a disk structure that is completely different from the normal structure for the Atari. This is done in the name of a standard file structure among p-systems on different computers. While it is possible to transport files in and out of the p-system, this is only possible through a special utility. There are two sorts of disks, or "volumes," as the p-system calls them. A TOS volume resides in a file on an ordinary disk. Native volumes take a whole disk and are slightly faster. A directory cannot be gotten from the desktop on a native disk, because the disk and GEM have absolutely nothing in common.

The p-system editor is anachronistic. Five years ago it may have been acceptable. But in the age of windows, it is difficult to believe that anyone could be happy with it. Using this editor is not at all intuitive. I would not recommend using it without studying the manual rather thoroughly. Since the p-system is separate from all other software, it is impossible to bring in a new editor from outside. With a little experience, the editor can be quite usable, but I will never be quite happy with it.

Everything about the p-system seems to be awkward. As an example: to get a directory of a volume, the filer must be invoked, the filer must be loaded from disk, a disk name given, and the filer quitted to return to the main menu. With all the disk accessing and typing that must be done, it can get quite tiresome.

The manuals recommend a two drive system. The p-system is very disk intensive -- I think that a

very large ramdisk or a hard disk would be more helpful. The p-system also seems to have some problems with running from a TOS volume: occasionally while swapping to disk it gets lost and must be rebooted. This does not happen with native volumes, but if a hard disk was used, this could be a problem.

A large amount of documentation was shipped with this system; however, quantity does not necessarily imply quality. It took me a week to figure out how to use a native disk. The majority of the documentation is oriented towards the IBM PC and the Pascal compiler. The graphics section has absolutely nothing to do with the Atari. Some documentation for the graphics was later sent to me on disk.

Graphics under the p-system are relatively primitive. The VDI and AES routines are completely unsupported. What is supported are turtle graphics. Turtle graphics can do quite a bit, but they do have limits.

The ANSI standard for FORTRAN 77 is really two standards: the minimal implementation, and the full implementation. The p-system FORTRAN is a minimal implementation. There are a few minor variations from the standard, but they really don't matter. The extentions include a statement to include a source file in a compilation or include code from a library. Minimal in this case is not a bad thing: the only thing I missed in this compiler was the string concatenation operator ("//").

Prospero FORTRAN

Prospero Software
LANGUAGES FOR MICROCOMPUTER PROFESSIONALS

It strikes me that I don't have nearly as much to say about this compiler. Prospero's FORTRAN is similar to a conventional mainframe FORTRAN system. It comes on a single disk that contains configuration programs, library maintenance programs, the GST linker, and, of course, the compiler. The first thing I noticed was that they provided no editor -- no big deal, but slightly surprising. The system also requires an installed run-time library; this can help keep the program files short, but it can also be a nuisance.

The Prospero compiler is a full implementation of the ANSI standard. A full GEM library patterned after the C library is also included. This setup works very well and lives up to Prospero's claims about serious business software.

Although I have done no thorough benchmarking, Prospero FORTRAN produces code much smaller and somewhat more efficient than Lattice C, the only compiler I have with which to compare it. With floating point numbers, it is over four times as fast.

There are a few problems with Prospero FORTRAN, however. I have found a few bugs, the majority of which are in the GEM library. Prospero has spoken to me and claimed that they have completely tested it and found no problems. I have recently sent a letter to them detailing the errors I have found. I think that the problems may stem from a recent discovery that British ST's (Prospero is British) are slightly different from the American variety. This has caused one other company, at least, a little discomfort. I will try to keep everyone up to date on this problem and its resolution, through my regular column.

There may be one other problem with Prospero: the run-time library. This little piece of software must be run once to install itself on a computer. It contains all of the necessary functions left out of every program to make it smaller. If saleable programs are going to be written in FORTRAN on this machine, the run-time library must go along. It is not mentioned in any of the licensing agreements, so by implication it is illegal to distribute it. I am trying to find out what exactly Prospero intended. Once again, I will keep you informed about any changes as I find out about them.

Conclusions

I can't say either product is perfect. I think both have a good purpose. Prospero's compiler is quite usable even with its small problems. I think it is more adequate than Pecan's, for business and commercial applications. Pecan's FORTRAN, on the other hand, would be perfect for someone trying to learn FORTRAN in an integerated environment. The only thing standing in its way is the p-system, which can be learned with a couple of weeks of concentration.

by Mark Lovell

Leader Board

Access Software Inc.
2561 S. 1560 W.
Woods Cross, Utah 84087
(801) 298-9077
($39.95)

Being an avid golfer, I was very interested when I heard that there several golf simulations that were coming out for the ST. Particularly, I was very happy to hear that Leader Board would be out.

I had seen it on the Commodore 64 and told the store owner that I wanted a copy when it came in for the ST. Well, it is here and it is even better than the version I had seen before.

The program is different from many golf simulators in that you view the course from behind the golfer instead of from above. Also, the graphics are superb. To a degree you have a sense of depth, which allows you to decide on which club to use rather than just depending on the distances. There is even animation involved in the swing, so that you can see the entire shot. It even has the little touches that make a good program enjoyable. These touches include the golfer looking up after the swing, so he can see where the shot went.

The program allows you to choose the level of difficulty, the course to play and how many holes to play. These combinations will permit you to play up to four consecutive rounds on four different courses with up to four different players at any of three levels of difficulty. The nice thing is the ability to mix and match these choices so that people of different skill levels can play a round together and be competitive. In addition to these choices, Access Software has announced that they will be producing additional courses for use with the program. These courses will come four to a disk at a price of $19.95. Unfortunately, none of the courses I've seen so far are designed to simulate any of the great courses of the world and the program doesn't allow you to design your own courses.

The program does allow you to choose which club to use and the manual that comes with it gives you some idea of the minimum and maximum distances each club will send the ball. But even that is not certain, because at the amateur and professional difficulty levels, the ball will hook or slice if you don't hit it properly. Also, at the professional level, you have to allow for the wind. In addition to all of that, you still have to contend with shots not traveling as far when you are hitting out of the rough or a sand trap. In this regard Leader Board presents you with all the decisions that you have to make on a real golf course. It even includes the need to "read" the break when you are putting.

There are some things that are not true to life in Leader Board. First, if you hit into water, you have to hit from where you started, but you are not charged a penalty stroke. In real life, you can drop a ball where your first one crossed the edge of the water, but you must count an extra stroke as a penalty. Secondly, the putting is unrealistic in terms of the number of long putts that you will make. In real life, even a pro won't make very many 30 foot putts. In this simulation, you'll probably make three or four in each 18 holes. Also, the short shots that you need to make around the green and out of sand traps are very unpredictable compared to real life, but then on the golf course most things are unpredictable.

When starting Leader Board most input is through the keyboard, but once you actually start to play everything is done with the mouse. There are keyboard commands to skip holes, go to the "driving range" for practice or even to start over. On the course, you use the mouse to select which club to hit, to start and end the swing and to indicate that you want to go on to the next shot. One area that caused me some problems at first was to remember to change clubs between shots. On the tee you will always have the driver in your hands, even if the hole is very short. If you don't change clubs, I will guarantee that you'll have an interesting shot coming back. Other than the automatic selection of the driver on the tee, the only club that is automatic is the putter. Once you have changed clubs, the club in your hands will stay the same until you change it or get within 64 feet of the hole. Once you get that close, you will be putting.

The program is copy protected, but does allow you to make back-ups. This is done by using a security key. The key is a plug that fits into the second joystick port. If this plug is not there, the copy of the program will not run. Unfortunately, I know that the first few packages received locally had some problems with the security key not fitting tightly enough, so the program did not recognize it was there. Also, I found that with my oldest ST the keyboard would not respond when the security key is in place. I did not have a problem with my other computer, but on the old one I could not type with the key in place, although the mouse did work. I have been unable to determine if the problem is a defect in the computer or a some change that may have been made in the hardware since my first ST was manufactured.

Overall, I am very impressed with this program. The simulation is very accurate in most respects. The different courses provide enough variety and difficulty to keep things interesting. The majority of what I have listed as problems I find are minor, but things that would be nice if they were changed or added. I have also seen people who are not interested in golf pick up this program and end up "golf nuts". I recommend Leader Board to anyone who enjoys simulations. I think you'll find this one both interesting and a challenge. In fact, a local computer store is considering holding a Leader Board tournament with gift certificates for prizes. Pardon me while I go practice.

Review by Greg Varley

PERSONAL MONEY MANAGER

Michtron, Inc.
576 S. Telegraph
Pontiac, MI 48053
($49.95)

"Personal Money Manager" is an accounting system which uses standard double-entry bookkeeping to record income and expenses. It also allows comparisons of actual and budgeted figures and will print reports.

The functions of the program for saving information are: **Account Maintenance**, through which you set up your annual budget in up to 999 income or expense accounts; **Enter Transactions**, which records income and expenses against the budgeted accounts in checkbook-style format; **Ledger Query**, which lets you review or change transactions; and **Automatic Posting**, which allows you to create multiple automatic transactions, such as dividing a paycheck into several expense accounts. The reports the program generates are: Annual Budget, which prints a month-by-month spreadsheet of the income and expense items; Expense Ledger, which prints a month-by-month spreadsheet for all income and expense transactions for the year; Account Summary, which prints a list of all the accounts with each account's current balance and year-to-date budget; Monthly Detail, which prints a list of all transactions by account during the month; and Single Account, which prints the transactions for the month for a single account.

"Personal Money Manager" is very easy to use and requires little reference to the manual, which is very short, after the first read-through. It is designed to save data automatically into the various files it uses. However, you must be sure to ask to save the data or it will be lost when you quit.

The program stores records into separate monthly files. Most of the reports generated will only print for the current month, as well. To change to a different month, you must use the Change Date function (or job, as it's called). Be sure to save your data before using Change Date. The program reminds you by asking if you want to save data when you quit the Enter Transactions, Ledger Query, or Change Date jobs.

Reports are printed directly to the printer. They cannot be looked at on the screen first. I consider this a drawback as I really don't want to go through printing copies on paper of all the reports just to find the one I want when I can't remember what they look like. A print-to-screen choice would enhance the value of the program tremendously.

To start using "Personal Money Manager," the budget accounts must be set up from the Account Maintenance job. If you try to enter a transaction to an account you don't have, you must cancel the transaction, return to Account Maintenance, and add the budget account before going on.

There is no quick way to repeat the same amount in each month of the budget account. The program will repeat the previous account's amount in the corresponding month of the new account. This made no sense to me and I found it tedious to have to enter the same figures in each month while setting up my budget accounts.

The program uses the mouse only for the pull-down menus. The rest of the cursor movement commands are keyboard entry only. The HELP key will cycle through all of the allowed responses, such as account number or account type, which is useful. The lack of "mouse-ability" in editing is a major flaw to me. I've seen recent ads which claim the program will also print checks, but this feature was not available on the version I reviewed. I don't find check-printing a necessary or desirable feature anyway, especially for home use.

All in all, "Personal Money Manager" does what it says it will do. It is also very simple to learn and use. I did not find it as useful for my needs as a checkbook program would be. It also doesn't do enough to be as useful to a business, even a small business, as a spreadsheet or general ledger program. It can only help you see where your money is going. If that is what you need, then "Personal Money Manager," particularly considering its simplicity and price, is a good program.

Reviewed by Mary Varley

Multi-FORTH
from **Creative Solutions, Inc.**
4701 Randolph Road, Suite 12
Rockville, MD 20852
(301) 984-0262

Multi-FORTH is fit for almost everyone. Almost every feature that gives other ST Forths their special appeal can be found on a file contained on one of the two Multi-FORTH disks. Everything. The turnkey fee, of course, is included in the system, and there are no upgrades. As I have said, everything is there. Except for a little skimping on the documentation, nothing seems to be missing. I have only one major hang up with Multi-FORTH, and that is its $150 price. For $150, you do not get a Forth that is worth $50 more than Level 1 4x FORTH,

and definitely not worth $100 more than Forthmacs or Mach 2. Aside from the price, there are many reasons to recommend Multi-FORTH.

From AES to XBIOS, Multi-FORTH has all the ST features covered. This includes dosound support, a multi level decompiler, local variables, and both a block and stream file editor. Although Multi-FORTH was developed with stream files, a block editor is provided to help port programs from both 4x FORTH and the now defunct H&D Forth. I have absolutely no complaints with the Multi-FORTH stream file editor, which is MicroEmacs. I don't particulary like the version of MicroEmacs supplied with Multi-FORTH (which is the same MicroEmacs that 4x FORTH users use), but then again, there are no better text editors. The block editor is the best ST block editor yet, although it falls quite short of MicroEmacs in both power and flexibility.

Both VDI and AES are fully supported, and a window handler is provided to ease the task of creating and manipulating windows. Despite the complete implementation of AES, Multi-FORTH does not allow the user to create his own resource files, which means that customized dialog boxes and menu bars cannot really be accessed until version 1.1. As with Mach 2, Multi-FORTH provides a vaporous explanation of VDI and AES. If you don't know what they are, the Multi-FORTH manual is going to be of no help.

There are three features of Multi-FORTH that merit particular distinction. Although these features have been implemented in other Forths, Multi-FORTH is the only Forth that provides all three. The first feature is the decompiler. A decompiler enables the user see how code was produced and greatly simplifies the task of programming. Local variables, explained in the Mach 2 review, are also part of Multi-FORTH. This makes programming math intensive applications far easier. The final feature of Multi-FORTH does not quite work, but is fun and interesting in its own way. With the word "RUN," the user can load up programs from either the ramdrive or the floppy and EXECUTE THEM! This means that you can load up a terminal program on top of Multi-FORTH, play Megaroids during if you are bored, or even load up ST Writer and jot down a poem. This feature was once implemented on Forthmacs, but it did not work very well with a good number of programs. Although the Multi-FORTH "RUN" commands works better than the version Forthmacs had, Multi-FORTH still cannot "RUN" Neo, the bouncing ball, or other programs. Despite the fact that Neo and the bouncing ball can be double clicked from medium res and work, Multi-FORTH cannot seem to run them. This is a minor griping point, as the sheer ability to load programs on top of Multi-FORTH is impressive enough.

The weakest link of the strong Multi-FORTH system is its documentation. The three ring Multi-FORTH binder lacks documentation for the demos, files that enhance the Multi-FORTH program, and the ST internals supported by Multi-FORTH. Despite its shortcomings, the documentation serves as an excellent Forth introduction and tutorial. The Multi-FORTH manual also provides excellent reasons for programming in Forth that every Forth programmer should know. The graphics are cheerful and unintimidating, as is the style of writing, which should satisfy everyone from the novice to the most hardened Forth addict.

Programming in Multi-FORTH very enjoyable, and is an experience to behold. Everything that you know how to do can be done. There is no excuse not to exploit your potential to its fullest with this Forth system. The editor is wonderful, debugging is a snap with the decompiler, and the Forth itself leaves nothing to be desired other than a little more zip. Multi-FORTH is by no means a slouch, but it cannot keep up with 4x FORTH or Mach 2 either. Sure, there are some bugs in the demos and the "RUN" feature will not run programs correctly sometimes, but Multi-FORTH has very few bugs in the system itself. The manual contains even fewer mistakes, with the most apparent being the two periods after "Rockville, Md" on the cover. The reason for this state of near perfection is the error bounty; Creative Solutions, which makes Multi-FORTH, will pay 50 cents for every typo you find, and $5.00 for every software error you find, provided you find it first. The bugs that I have found have in most likelihood already been reported, and will be fixed with revision 1.1. With an increase in speed and a price drop to $50, Multi-FORTH version 1.1 could stage a coup on the rest of the ST Forth systems. But as it is, Multi-FORTH still costs $150, and all you can get is a minutely flawed, powerful version 1.0.

***

Reviewed by Albert Lew

Mach 2: Flying at high quality speed.
Palo Alto Shipping
P.O. Box 7430
Menlo Park, CA 94026
(415) 854-2749

Without question, Mach 2 is a winner. It is sleek, fast, and loaded with all the latest high tech gadgets. Even the operator's guide invokes visions of elegant, high speed operation. But Mach 2 is not a fighter plane; it is a "Forth

Development System for the Atari/ST" produced by the Palo Alto Shipping Company. Like all high performance machines, Mach 2 has to make some compromises, but when the disadvantages are weighed against the advantages, it is clear that Mach 2 presents a fresh vision in the Atari ST Forth world.

My favorite part of the entire system is the documentation. Mach 2's documentation does the best job of all the ST Forths in presenting a manual that is very understandable to both beginners and experts alike. Many topics I have not yet covered in A Step Forth are explained concisely and clearly with the help of excellent diagrams and pictures. Like every ST Forth save Forthmacs, Mach 2's documentation was produced using a Macintosh, but interestingly enough, the typeface used with the Mach 2 manual is far easier to read than that of the other manuals, although the pitch size appears to be about the same. The glossary is perfect for the beginner because it precisely defines what each word does and how to use it. For experienced hands, the format of listing one word per page in the glossary will make word lookup far easier. The one shortcoming of the documentation is that it makes only a passing reference at GEM. Windows are covered at length, but without much reference to the ST's window handler, AES. On most Forths, this would be acceptable, but because Mach 2 describes everything else from the ground up, I see no reason not to include a discussion of fulling exploiting the ST hardware as well.

Of course, all links to ST hardware within the program itself are there. GEM in all its glory is supported, as well as all the BIOS and XBIOS commands. Basically, this means that you can manipulate MIDI, the serial port, windows, VDI primitives, sound, and other ST features to their full potential. Many good examples are provided on the Mach 2 disk to show how to fully exploit certain features of the Mach 2 Forth system. The system itself is run under GEM, and as I found out, that is both an advantage and a disadvantage. With the other ST Forths, loading files for editing falls just short of being intuitive. With Mach 2, however, files can be loaded by moving the mouse up to the "File" word on the menu bar and clicking the appropriate selections. Using GEM also allows the user to utilize any desk accessories at his disposal. With the other Forths, learning a new Forth techinque via Compuserve and then immediately switching to the Forth program to try it out would be impossible. Of course, it is child's play with Mach 2. Interestingly enough, Mach 2 doesn't seem to lose any speed as a result of running under GEM. If anything, it is in the same league as version 3 of 4x Forth (the speed champion), and a good deal faster than the other ST Forths I have seen.

The main disadvantage of running Mach 2 under a GEM environment is the editor. Personally, after having used MicroEmacs to edit programs for 4x Forth, Forthmacs, and Multi-Forth, I found the Mach 2 editor to be both underpowered and slow. Since Mach 2 does use stream files, however, this problem can be overcome by leaving Forth and loading up MicroEmacs. Of course, this method loses the immediacy between editing and execution that is one of the main strengths of any Forth programming environment, but I felt that I simply could not live with the Mach 2 editor. After talking to the people at the Palo Alto Shipping Company, however, I learned that many people have voiced similar complaints and a revamped editor is already in the works. The other picky point about a GEM-based Forth is the cursor. In Mach 2, the cursor is an underline ("_"), which does not work properly all the time and is much harder to spot than the normal blinking rectangle. It is very annoying to type in a line of text in Mach 2 only to have the bottom scan line of the text come out as garbage. Despite these shortcomings, it is definitely refreshing to have GEM when running the Forth and more of a help than a hinderance.

A splendid feature of Mach 2 is its ability to handle local variables. Local variables make math in Forth far easier to use, and are my next favorite feature in Forth next to the decompiler. With local variables, math is performed more along the lines of traditional languages, instead of with the stack. The stack method is fine for people who are accustomed to it, but for those whose are familiar with algebra, and not Forth, local variables are more natural than the stack.

Despite my many criticisms of Mach 2, I still love it. It lacks many of the disk access features of Multi-FORTH and Forthmacs, and the dictionary is one of the smallest I've seen; 4x Forth dwarfs it. Mach 2 takes up about 300K of memory, which is a little too much for a 512K system. The techinal description of how Mach 2 works will not satisfy many, but the whole Forth is written in assembly, which should allow those who are really interested to figure out exactly what goes on behind the scenes. Without question, Mach 2 is the system I would recommend most to users who are not familiar with Forth and want to learn it painlessly or users already know Forth and want to painlessly manipulate windows. For the Forth and ST hardware sophisticate, I would dispense one tepid warning about Mach 2 before recommending it. Mach 2's weakest link is its editor, but if Palo Alto Shipping Company has its way, the new Mach 2 editor will quiet all the complaints. Local variable support, a decompiler, GEM environment, speed, and

no distribution fee -- Mach 2 would be a great deal even if it cost $200. Amazingly, a $59 check to the Palo Alto Shipping Company, P.O. Box 7430, Menlo Park, CA, 94026, is all that stands between you and Mach 2. (CA residents add 6.5% sales tax.) Any sales inquiries can be made to 800-44FORTH. **** by Albert Lew

---

Business Statistics
Experimental Statistics
Forecasting and Time-Series
Sales and Market Forecasting
($145 Each)

Lionheart
P.O. Box 379
Alburg, VT 05440
(514) 933-4918

To bridge the transition between the game machine concept for the ATARI and the business machine, strong analytical programs are needed. Lionheart is known in the MS-DOS and Apple world for having "heavy duty" numerical analysis programs that use minimal graphics. These four programs are the first that have been translated onto the ATARI format. In general, they are usable, fast, reliable programs that fulfill a very specific need. They work well and easily. There is good support from the author. I can recommend them with no qualms.

Let me generally explain the good points of the programs. First, they are quite fast. Performing the calculations takes seconds, rather than minutes if they were uncompiled. The data is generated and stored as ASCII data files, which make them easy to be read by word processors and changed and modified. The data can be accepted by VIP Professional to be used by its graphics package. VIP Professional can also generate the data. There are many, many checks and data modifications allowed. The programs are well debugged.

Each package comes with both an instruction book on how to use the statistics as well as how and what each individual program does. The books include many examples and illustrations. They are useful for brushing up on your knowledge or approaching the subject for the first time.

The disks are not copy protected and there is even short snippets of BASIC code given to show how to enter data to the various programs.

The programs are discreet. They allow you to enter the data either from a disk file or from the keyboard with the opportunity to save to disk. The printer is supported and the majority of the data that is calculated can be sent to be printer for a hard copy.

There are a few areas in which I would have liked to see some improvements. These programs are written in BASIC and ported over from an MS-DOS system and then compiled. There are little error trapping that explains what was done wrong. An error throws you back to the GEM tabletop. Since it is a compiled BASIC, some of the timing loops are too fast to read all of the messages that might be generated. There are no real graphics capabilities in any of these programs. The programs leave much to be desired in terms of user friendliness as shown by other programs for the Atari ST. There is no GEM, no mouse, and no colors.

As with all the packages, there is an excellent manual supplied with the programs. The experimental statistics book explains many subjects including the topics of uncertainities, sampling distributions, the application of the normal distribution, hypothesis testing, significance levels and the calculation of Type I and Type II errors, the analysis of variance, linear and multilinear regression. The book is over 250 pages and there are 44 programs associated with the package.

Some of the programs in the experimental statistics package perform the following operations. DIST calculates the probabilities for normal, chi-square, t, and F test, so that tables are no longer needed. GRAPH plots up to three sets of data together. HYPOTEST contains hypothesis tests for comparison of means and variances. MREG calculates multilinear regressions with up to nine independent variables. MVREG performs multivariate regression with an infinite number of variables. ORTHOREG uses orthogonal polynomials t`up to the sixth order. TABULATE cross tabulates correlations, contingency analysis, and the analysis of frequencies of response. There are three programs for analysis for one-way, two-way, and three way factorial analysis of variance. There are programs to perform the analysis of variance for Latin Squares, Graeco-Latin, and Youden square analysis. There are programs for the analysis of variance and regression to first 2-, 3-, and 4- dimensional orthogonal response surfaces. SMOOTH and SORT manipulate the data. There is even a program to generate random telephone numbers.

The business statistics set manual shows how to characterize performance by comparison to means or variances. It shows how to use quality control methods for product reliability. There are chapters on parametric and non-parametric

significance testing. Finally there are several chapters on how to design business experiments. There are 42 programs associated with this package.

There are some overlapping in the programs with the Experimental statistics package. Such programs include DIST, HYPOTEST, MREG, MVREG, SMOOTH, SORT, and TABULATE. There are also additional programs. CONTINGE is used to generate RxC contingency tables. SAMPLE shows how to perform stratified sampling, cluster sampling, sequential, same, binomial and hypergeometric schemes. PERIOD analyizes the periodicity of seasonal and cyclic time series. TREND removes such variations. MAVFCAST and EXPFCAST performs forecasting by moving average and exponential smoothing techniques. AUTOREG performs the autoregression analysis of a time series.

The Forecasting and Time Series package contains a 150+ page manual which explains the methodology of analyzing time series of data. The methods include correlation, autocorrelation, and spectrum analysis using Fast Fourier transforms. The forecasting methods include moving average, exponential smoothing, regression, and Box-Jenkins ARIMA type models. To help in the understanding of these methods, examples are given and fully explained. The data on the disk includes Dow-Jones utilities index information as well as other real data to help learn how to use the programs. There are 27 programs with this package.

The programs again contain a slight amount of overlap with GRAPH, MREG, and PERIOD found in the experiemental statistics package. The other programs include FFT for Fast Fourier transforms for spectrum analysis. TRANSREG which calculates relationships between two time series into terms of a transfer function. Three programs MAVFCAST, EXPFCAST, BJKCAST which forecast by either moving averages, single or double exponential smoothing or three parameter Box-Jenkins techniques. ARIMA uses an advanced Box-Jenkins technique.

Sales and Market Forecasting is a companion program to the other forecasting programs. The manual is 150+ pages and the emphasis is more on marketing forecasting. The manual shows how to adjust data for inflation and population as well as seasonal characteristics. It shows how to discover trends and cycles. It evens explains how to get the necessary data to use for correlations. There are 32 programs with this package.

This package has the most overlap. All the programs from the Forecasting and Times series package are included. The new programs include LREG which fits one time series up to three other lagged time series. ADSERIES and NORMILIACR are utility programs that allow for the addition and subtraction of series as well as the multiplication and division of one series by another. An additional series of data files are included to help in learning how to use this package.

The pricing of the four packages shows the overlap involved in the programs. Business Statistics and Experimental statistics each list for $145. The combination package for these two is $200. The Forecasting and Sales and Marketing packages are $145 and $150 respectively. The combination package is also $200. The difference in cost is partially due to the programs, but an additional manual is also supplied.

What is the bottom line on these packages? They are expensive comprehensive programs. The manuals are used to explain the underlying principles of all of the programs and succeed very well. The manuals are well written, but they assume some knowledge of statistics. These programs are designed for no nonsense statistical analysis. They run well and produce correct answers. They will take a bit of study in order to use well. For the number of programs that each package gives, and the amount of usage that each will receive, these programs are a good solid value. They fill a niche and should be purchased by any one that needs good statistical packages.

If I ever go back to teaching Engineering, I will certainly use these programs as examples.

Reviewed by Sol Guber

Advertisers Index

25	Antic Software
47	Artworx Software
54	BayView Software
57	Cal Com Inc.
17	Central Point Software
BC	Computer Palace
IBC	Computer Software Services
47	Forthware
IFC	Insoft
57	Logical Design Works, Inc.
45	Palo Alto Shipping
53	Proco Products
13	TDI Software, Inc.
3	Thoughtspace Development

Our Dealers

* V.G. Data Shack
Brossard, Que. J4Z 3G3 Canada
(514) 445-9598

* Ali Computers
Gloucester, Ont. K1J 8V1 Canada
(613) 744-0220

Next Step Computers
Winnipeg, Man. R2V 3C8 Canada
(204) 632-8602

* Gemini Sales, Inc.
Burnaby, BC V5C 2K7 Canada
(604) 294-9717

* Family Computing Ltd
Fort St. John, BC Z1J 3V5 Canada
(604) 787-9063

Compuclub
Natick, MA 01760
(800) 692-8274

Freeport Systems
Quincy, MA 02169
(617) 472-6139

* Software Haus
Quincy, MA 02170
(617) 770-3899

The Bit Bucket
West Newton, MA 02165
(617) 964-3080

* Software Connections
Warwick, RI 02886
(401) 738-3430

Hands On Computers
Westbrook, ME 04092
(207) 854-1155

* Comptec
Wallingford. CT 06492
(203) 237-3929

* Comstat
Wallingford, CT 06492
(203) 284-0548

* Derric Electronics
Hamden, CT 06517
(203) 248-7227

* Software Spectrum
North Plainfield, NJ 07060
(201) 561-8777

* Software City
Teaneck, NJ 07666
(201) 692-1879

* Gemini Enterprises
Cedar Knolls, NJ 07927
(201) 267-0988

Computer System Consultants
Hightstown, NJ 08520
(609) 448-8888

Micro Con
Lawrenceville, NJ 08648
(609) 799-6565

Village Computer, Inc.
New York, NY 10012
(212) 254-9191

Leigh's Computers
New York, NY 10028
(212) 879-6257

Computer Software Plus
Brooklyn, NY 11229
(718) 645-1880

Great American Software Store
Flushing, NY 11358
(718) 357-5522

Computerware
East Meadow, NY 11554
(516) 731-7939

Little Computer Shop
Centereach, NY 11720
(516) 467-4352

* Computer Palace
Selden, NY 11784
(516) 698-6183

* Computer Alternatives
Clifton Park, NY 12065
(518) 383-2898

Software City
Syracuse, NY 13224
(315) 445-2577

* Video Challenge
Tonawanda, NY 14150
(716) 837-2611

Buffalo Computer Center
Buffalo, NY 14226
(716) 835-0648

The Software House
Fairport, NY 14450
(716) 223-7658

* Personal Computers of Ithaca
Ithaca, NY 14850
(607) 272-2831

* Computers & Games
Reading, PA 19605
(215) 929-0540

The Program Store
Washington, D.C. 20024
(202) 863-1947

Applied Computers, Inc.
Gaithersburg, MD 20877
(800) 428-2747

The Program Store
Kensington, MD 20895
(301) 984-1233

Cal Com, Inc.
Wheaton, MD 20902
(301) 681-9121

Compuvision
Millersville, MD 21108
(301) 987-0998

Home Computers
Towson, MD 21204
(301) 337-2733

Diskovery
Falls Church, VA 22044
(703) 536-5040

L & Y Electronics
Woodbridge, VA 22191
(703) 494-3444

IO Computers
Norfolk, VA 23505
(804) 583-1609

* TDS Computers
Carrboro, NC 27510
(919) 929-4593

* TDS Computers
Durham, NC 27705
(919) 286-3775

* Computers Mean Business
Tucker, GA 30084
(404) 934-5295

* Rainbow Computer Center
Orlando, FL 32803
(305) 894-2270

Future Elecronics
St. Petersburg, FL 33703
(813) 527-7543

Computer Logic
St. Petersburg, FL 33709
(813) 546-3137

The Computer Chip
Sarasota, FL 34243
(813) 355-8448

VideoVisions
Louisville, KY 40222
(502) 425-2022

Reality World Computers
Lexington, KY 40503
(606) 223-5274

* Program Store
Columbus, OH 43214
(614) 457-1153

Video Express
Columbus, OH 43227
(614) 866-2685

* Video Computer World, Inc.
Oregon, OH 43616
(419) 691-7282

* Futuretronics
Elyria, OH 44035
(216) 322-2831

Kristy Computers
Elyria, OH 44035
(216) 324-2240

B & G Electronics
Lakewood, OH 44107
(216) 521-2855

* Ram-Run Computer Products
Medina, OH 44256
(216) 723-8929

* Microwave Magic
Fairfield, OH 45014
(513) 874-6560

Electronic Connexion
Centerville, OH 45459
(513) 435-8956

Microcomputers, Inc.
Indianapolis, IN 46224
(317) 291-8882

* Computer Corner
Merrillville, IN 46410
(219) 738-3282

* Digital Deli
South Bend, IN 46635
(219) 277-5026

Don's Computers
West Lafayette, IN 47906
(317) 743-4041

*Canton Computer
Canton, MI 48187
(313) 459-4340

Kamtech Electronics Svc.
Appleton, WI 54915
(414) 739-8477

Wizard's Word
New Hope, MN 55427
(612) 545-2136

* User Friendly Computers
Spring Lake Park, MN 55432
(612) 786-8181

* Software Plus
Wheeling, IL 60090
(312) 520-1717

* DigitalWorld
Addison, IL 60101
(312) 543-9000

* Software Plus West
Hanover Park, IL 60103
(312) 837-6900

* Mars Merchandizing
Elmhurst, IL 60126
(312) 530-0988

* JAF Data Systems
Chicago, IL 60643
(312) 238-4348

* Coz Enterprises
Chicago, IL 60647
(312) 235-9629

Hobby Town
Lincoln, NB 68505
(402) 464-2858

* Software Center International
Metairie, LA 70002
(504) 885-2000

* American Networks
Marrero, LA 70072
(504) 341-7222

Computers Plus
Jacksonville, AR 72076
(501) 982-2986

Baker Value Rite
Little Rock, AR 72203
(501) 565-0336

Video-Computer
Lawton, OK 73505
(405) 355-9798

Computer Discovery
Dallas, TX 75244
(214) 484-9104

* Computer Skills
Euless, TX 76040
(817) 267-5151

* The Floppy Wizard
Houston, TX 77024
(713) 461-8660

* Computers to Grow
Houston, TX 77074
(713) 777-1673

Mindwares Scientific
San Antonio, TX 78216
(512) 340-1471

Peoples Computers
San Antonio, TX 78229
(512) 690-9246

* Jenkins Computer Store
El Paso, TX 79924
(915) 751-6938

The Computer Room
Aurora, CO 80014
(303) 696-8973

The Computer Room
Englewood, CO 80111
(303) 799-9733

* Horizon Computers
Denver, CO 80210
(303) 777-8080

* Rocky Mountain Atari Svc.
Boulder, CO 80301
(800) 662-8274

Computer Link
Boulder, CO 80302
(303) 444-7300

Citadel Computers
Colorado Springs, CO 80909
(303) 591-1700

* ZCMI, Pineridge
Chubbuck, ID 83202
(208) 239-7144

* ZCMI, Grand Titon
Idaho Falls, ID 83401
(208) 525-4075

Digital Doohickeys
Idaho Falls, ID 83402
(208) 529-5830

ABI Computer Video
Nampa, ID 83651
(208) 465-7515

* ZCMI, Laton Hills
Laton, UT 84041
(801) 546-5039

* ZCMI, University
Orem, UT 84057
(801) 227-3031

Lloyd's Computers
Orem, UT 84058
(801) 225-5751

* ZCMI, South Towne
Sandy, UT 84070
(801) 321-6846

Software Hut
Salt Lake City, UT 84111
(801) 355-0066

* ZCMI, Downtown
Salt Lake City, UT 84111
(801) 321-6000

Computer Technica
Salt Lake City, UT 84115
(801) 485-2628

* ZCMI, Cottonwood
Salt Lake City, UT 84117
(801) 321-6284

* ZCMI, Valley Fair
West Valley City, UT 84119
(801) 321-6708

Digital World
West Valley City, UT 84120
(801) 966-6148

* ZCMI, Cache Valley
Logan, UT 84321
(801) 750-7510

* ZCMI, Ogden
Ogden, UT 84401
(801) 778-2109

Micro Mania
Ogden, UT 84403
(801) 479-0500

* Bits "N" Bytes
St. George, UT 84770
(801) 628-5755

* ZCMI, St. George
St. George, UT 84770
(801) 628-5291

Computer Works
Glendale, AZ 85301
(602) 842-1341

Star Type
Glendale, AZ 85302
(602) 934-7387

* Bookmans
Tucson, AZ 85716
(602) 325-5055

Wiser Electronics
Las Vegas, NV 89104
(702) 385-7782

* Computer World
Las Vegas, NV 89109
(702) 796-1377

* Cimarron Computers
Sparks, NV 89431
(702) 331-8020

* Computer House
Sparks, NV 89431
(702) 356-7216

Opamp Technical Books
Los Angeles, CA 90038
(213) 464-4322

Lamar Micro
Redondo Beach, CA 90278
(213) 374-1673

* Computer Place
Torrance, CA 90505
(213) 325-4754

* Cal Com Inc.
Buena Park, CA 90620
(714) 523-5353

* Mid-Cities Comp-Soft
Bellflower, CA 90706
(213) 867-0626

Personal Computers, Etc.
Canoga Park, CA 91303
(818) 883-5421

ComSoft
Reseda, CA 91335
(818) 768-5017

* Atcom Micro Center
Thousand Oaks, CA 91362
(805) 497-1220

* Warner Engineering
San Diego, CA 92110
(619) 294-4024

DiskBelief
San Diego, CA 92123
(619) 565-1078

Pro-Data
San Diego, CA 92154
(619) 423-3442

* Nova-Electronics & Software
Riverside, CA 92501
(714) 781-7332

Learning Tree Computer Center
Santa Ana, CA 92705
(714) 667-1575

* Consumer Electronic Store
Anaheim, CA 92805
(714) 635-8621

* Paradise Computer Systems
San Luis Obispo, CA 93401
(805) 544-7127

Information Systems Unltd
Santa Maria, CA 93454
(805) 922-4545

Alpha Comp. & Fin. Svc.
Lancaster, CA 93534
(805) 942-2626

* Ridgecrest Computer Ctr.
Ridgecrest, CA 93555
(619) 375-4364

Computer Literacy Bookshop
Sunnyvale, CA 94086
(408) 730-9955

Fry's Electronics
Sunnyvale, CA 94086
(408) 733-1770

* Access to Software
San Francisco, CA 94118
(415) 751-2231

3E Software & Systems
Hayward, CA 94541
(415) 537-3637

Home Computers
San Leandro, CA 94578
(415) 278-8881

ECX Computer Company
Walnut Creek, CA 94596
(415) 944-9277

* Microworld
Berkeley, CA 94703
(415) 841-9179

* Winners Circle
Berkeley, CA 94704
(415) 845-4814

* Software 1st
San Rafael, CA 94901
(415) 459-1475

* B&C Computervisions
Santa Clara, CA 95051
(408) 749-1003

* San Jose Computer
San Jose, CA 95125
(408) 723-2025

* Empire Computers
Santa Rosa, CA 95401
(707) 526-4106

* Software 1st
Santa Rosa, CA 95401
(707) 576-0972

Sawyer's News, Inc.
Santa Rosa, CA 95404
(707) 542-1311

* Computertime
Citrus Heights, CA 95610
(916) 969-4111

Omni Computers
Chico, CA 95926
(916) 893-3458

Computronics
Honolulu, HI 96819
(808) 836-0273

High Tech Toys
Beaverton, OR 97005
(503) 646-3950

* IB Computers
Portland, OR 97225
(503) 297-8425

Univ. of Oregon Bookstore
Eugene, OR 97401
(503) 686-4331 ext. 54

* Computer Palace
Eugene, OR 97402
(503) 683-5361

Sunshine Computer Center
Medford, OR 97501
(503) 773-3608

Serious Software
Grants Pass, OR 97526
(503) 479-9516

Butler's Computer Service
Federal Way, WA 98003
(206) 941-9096

* The Computer Cache
Anchorage, AK 99501
(907) 272-9941

* Xanth Computer Systems Inc.
Seattle, WA 98104
(206) 624-9292

Computer Palace
Kenai, AK 99611
(907) 283-9088

We are also distributed by:

*** Midwest ***
Computer Software Services
Elk Grove Village, IL 60007
(800) 422-4912

*** Rocky Mtns & Plains ***
Horizon Computers
Denver, CO 80210
(303) 777-8080

*** Los Angeles Area ***
Levity Distributors
No. Hollywood, CA 91601
(818) 506-7958

ST Applications
THE ATARI 520/1040ST JOURNAL

We currently have over 300 retail outlets. One hundred thirty are ours directly with the remainder serviced through our three distributors.

This listing of our retail dealers is provided as a service to both our dealers and readers. Dealers are listed in zip code order. If you do not find your store listed here please drop us a line identifying your distributor and we will gladly add you to our list.

Should you not be able to find ST Applications at your local ST dealer Please drop us a line and we'll send them a sample copy.

Our only products are ST Applications and the monthly Support Disk.

ST User Groups

AA-AUG ST Sig
c/o Jay Mcarthy (714) 829-8722
Upland, CA

AACE ST Sig
c/o David Mann (512) 346-4940
7108 Spurlock Dr.
Austin, TX 78731

ABACUS ST Sig
c/o Warren Lorente (415) 453-3665
San Francisco, CA

Atari Computer Enthusiasts of
Columbus ACEC ST Sig
P.O. Box 30049
Gahanna, OH 43230
Lawrence Mendel, Pres.

Adelaide Atari Computer Club
Norman Pearce, Pub. Rel. Officer
P.O. Box 333
Norwood, S.A. 5067 Australia

Atari Central Users Group
c/o John Buckner (602) 881-0539
2429 N. Richey Blvd.
Tucson, AZ 85716

Atari Computer Owners of Rochester
Kathy Scoville (716) 334-5820
P.O. Box 855
Fairport, NY 14450

Atari Elite
c/o Stephen Kotula (412) 653-1641
P.O. Box 58
Library, PA 15129

Atari ST UG of the Palm Beaches
BBS (305) 793-9385
(300baud, 10pm-6am)

Austin ACE ST Sig
c/o Dave Mann (512) 346-4940
8207 Briarwood Lane
Austin, TX 78758

B.A.S.I.C. ST Sig
c/o Pete Fazio (718) 646-6384
2724 E. 23rd Street
Brooklyn, NY 11235

Blue Mtns Homebrew Computer Club
Eric Lindsay, Editor
6 Hillcrest Avenue
Faulconbridge NSW 2776
Australia

BRACE ST Sig
Tom Tjarnberg
P.O. Box 6341
Bellevue, WA 98008

Cin'tari, Inc. ST Sig
Neil Reitz (513) 574-8622
P.O. Box 14959
Cincinnati, OH 45214

COAST Users Group
Chuck Thorpe (415) 945-1949
1015 Esther Drive
Pleasant Hill, CA 94523

Desert GEMs
Ridgecrest, CA
(619) 375-4364.

FASTER
Andre Lafreniere, Librarian
1161 Des Fauvettes
Boucherville, Quebec
Canada J4B 6A8

FEAST Far East Atari ST
Dale Ellis
P.O. Box 7075
APO San Francisco, CA 96328

Greater New Haven ST UG
c/o Robert Fischer (203) 288-9599
80 Kildeer Road
Hamden, CT 06517

HASTE Houston Atari ST Enth.
c/o Patrick Mahoney
622 Cherrybark
Houston, TX 77079

High Sierra Atari Users' Group
c/o Gene Manson (702) 356-6252
P.O. Box 2152
Sparks, NV 89432

Indy ST Computer Club
c/o Jim Wallace (317) 898-9856
Box 2525
Indianapolis, IN 46206

J-BUG Jackintosh Boston UG
Alan Glick (617) 296-8286
One Center Plaza
Boston, Massachusetts 02108

LAACE (Los Angeles ACE) ST Sig
c/o Kent Simon (818) 349-7540
P.O. Box 7752
Van Nuys, CA 91409

Lea Valley Atari Users Group
Matt Tydeman, VP & Editor
125 Cadmore Lane
WALTHAM CROSS
Hertfordshire EN8 9JH England

MACE Michigan ACE ST Sig
c/o Fred Kandah (313) 665-8982
536 Elm St
Ann Arbor, MI 48104

Pittsburgh Atari Computer
Enthusiasts (PACE) ST Sig
c/o John Babson
P.O. Box 13435
Pittsburgh, PA 15243

Portland Atari Club (PAC) ST Sig.
P.O. Box 1692
Beaverton, OR 97005

Rocky Mountain Atari Advanced UG
c/o Myron Drapal, treasurer
313 W Lucerne Dr.
Lafayette, CO 80026

San Leandro Computer Club ST Sig
Bob Barton (415) 352-8118.
P.O. Box 1525
San Leandro, CA 94577-0152

Seattle Puget Sound ACE ST Sig
c/o Dave Showalter (206) 824-5141
P.O. Box 100576
Tacoma, WA 98411

ST Ace
c/o Mark Taylor
2173 W. Steele Ln.
Santa Rosa, CA 95401

STACK (ST Atari Computers Korea)
c/o Robert Mann
A Co 102d MI Bn
APO San Francisco, CA 96224-0373

STAG (ST Atari Group)
c/o John Huston
6320 Keeler Street
Huber Heights, OH 45424

ST Atari Road Runners
c/o Robert Hueffman (203) 421-3864
48 Winding Rd.
Madison, CT 06443

STING -ST INterest Group
8461 Plaza Blvd.
Minneaplois, Minn. 55432
612-786-8181

ST National Users Group (STNUG)
3650 Sand Court
Mims, FL 32754
BBS (305) 383-1413.

STORK ST Owners Representing Kadena
c/o Barry C. Watson
PSC #2 Box 14928
APO San Francisco, CA 96367

ST SOTA (State of the Art) UG
c/o Jeff Rigby (813) 922-6244
3949 Sawyer Rd.
Sarasota, FL 33583

STUGof Winnipeg
c/o Len Stokes
227 Kenaston Blvd.
Winnipeg, Manitoba, Canada R3N 1V5

STUN (ST Users Network)
c/o Dave Johnson (801) 964-0771
4308 S. 4625 W
West Valley City, UT 84120

Tampa Bay ST Users Group
c/o Future Electronics
St. Petersburg, FL
(813) 527-7543

Toronto Atari Federation ST Sig
c/o Jon Manol
Mailbox 1527
5647 Yonge Street
Willowdale, Ont M2M 4E9 Canada

ST Bulletin Boards Update

Here we present updates to our list of ST BBS's published last month. To the best of our knowledge the information in this list is accurate. However, we take no responsibility for their existence or operation. (as of 12 Sep 1986)

	Name	State	Number	Notes
	STUG BBS	Can	204 633-3066	ST Atari UG, Winnipeg
*	Starnet (Illiana ACE).	IL	217 267-3530	FoReM, 20Meg
*	Atari Base Line #3 . .	CA	408 745-2642	Atari Corp.,HD
*	Atari Base Line #4 . .	CA	408 745-4758	Atari Corp.,HD
*	Atari Base Line #5 . .	CA	408 745-5664	Atari Corp.,HD
*	STATE.	CA	707 585-2194	ST Applications

* indicates that we've logged on. All boards listed are 300/1200 baud.

Due to the intermittant nature of some BBS's, as well as inaccuracies of dialing please be courteous and listen to the ring & answer for the first time you attempt to log on to a new board.

New Software Etcetera

The following listing includes both last month's errors in pricing and new products we've found out about.

Absoft Corp
4268 N. Woodward
Royal Oak, MI 48072
(313) 549-7111

AC/Fortran ($195) from MAC files can easily be ported to the ST. ANSI 77 standard & IEEE floating points.

Access Software
2561 South 1560 West
Woods Cross, UT 84087
(801) 298-9077

* **Leader Board** ($39.95) is computer golf with excellent graphics.
* **Tournament Disk #1** offers 4 more courses and improved graphics.

Tenth Frame is a computer bowling simulation due out by Xmas.

Cosmi
415 N. Figueroa Street
Wilmington, CA 90744
(213) 835-9687

* **Super Huey** ($39.95) is a helicopter flight simulator.

Mad Scientist Software
2063 N. 820 W.
Pleasant Grove, UT 84062
(801) 785-3028

Cardiac Arrest ($69) is an adventure game for anyone who wants to play doctor.

Mark Williams Company
1430 West Wrightwood Avenue
Chicago, Illinois 60614
(312) 472-6659

* **Mark William's C** ($179.95) is a powerful, well received 'C' compiler.

Master Designer Software
5743 Corsa Avenue, Suite 215
Westlake Village, CA 91361
(818) 889-1537

Cinemaware distributed by Mindscape, is a collection of interactive movies with arcade sequences.

SDI (Strategic Defense Initiative) with 25 screens, is best with joystick (October 15th).

Defender of the Crown (as seen on the Amiga) will be out in January '87.

Megasoft
P.O. Box 1080
Battle Ground, WA 98604
(206) 687-7176

* **A-Copier** ($39.95) disk copy program
A-Filer/Report Writer ($59.95)
A-Mailer ($49.95)
Graphic Label maker ST ($39.95)
Ram Disk and Spooler ($39.95)
ST Tools ($39.95)
Telecommunications Package ($39.95)

Metacomco, Inc.
5353 #E Scott's Valley Drive
Scott's Valley, CA 95066
(408) 438-7201

Cambridge LISP ($199.95) -An interpreter/ compiler providing a complete LISP development environment.
* **Lattice C** ($149.95) is a full K&R implementation C compatible with Lattice compilers on the IBM PC etc.
* **Macro Assembler** (79.95) Professional quality development system with Standard M68000 mnemonics.
* **MCC Pascal** ($99.95) Fast ISO/ANSI standard compiler producing native 68000 code.
BCPL ($149.95) -Full standard BCPL compiler combining the convenience of high level language with assembler flexibility.
Metacomco MAKE ($69.95) - A UNIX acompatible MAKE utility with extended functions.
* **Menu+** ($29.95) -Powerful menu generator including batch mode, HISTORY, creation of tools, arguments & options, organization of working files, programs etc. Included with each Metacomco language.

Progressive Computer Applications, Inc.
2002 Mccauliffe Drive
Rockville, MD 20851
(301) 340-8398

* **Graphic Artist** Ver 1.5 ($199.95) is a combination of Computer Aided Design (CAD), business graphics, free-hand drawing, and typesetting programs in one package. Supports dot-matrix, color-, and laser-printers.
* **Font Editor** ($79.95), from the Graphic Artist package.

Softlab
168 N 100 E
St. George, UT 84770
(801) 628-5400

* **OSCAR** ($39.95) GEM-based hard disk backup utility which archives larger than 720k files to single disk. Also features file & directory restore.

Startroniks
P.O. Box 8185
Turnersville, NJ 08012
(516) 884-3196

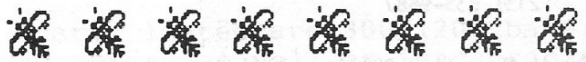
* **Font Writer Plus** ($39.95) adds multiple font printing and picture integration capabilities to your favorite ST word processor (ie. ST Writer, First Word, etc.). Works with your Epson compatible printers (including laser printers) to create distinctive newsletters and reports.

Supra Corp.
1133 Commercial way
Albany, OR 97321
(503) 967-9075

* **10 Meg. Harddisk** ($549)
* **20 Meg. Harddisk** ($799)
30 & 60 Meg Harddisks ($995 & $1995) are scheduled shortly.
* **Microstuffer** ($69.95) is a 64k printer buffer.
* **Omega Terminal** ($29.95)
* **Supra 300ST Modem** ($59.95) includes Omega Term.
Hard Disk Backup This unit should be avail. in Dec and will backup 20meg per tape.

Timeworks, Inc.
444 Lake Cook Road
Deerfield, IL 60015
(312) 948-9200

Word Writer ST with Spell Checker ($89.95)
Data Manager ST with Report Writer ($89.95)
Swiftcalc ST with Sideways ($89.95)

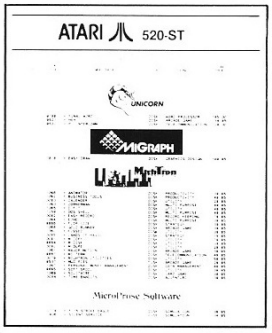


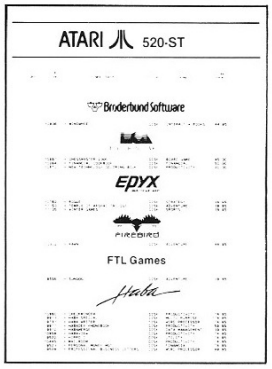
It's the "Bible" of the industry, over 300 pages crammed with information, 1½" thick, over 2 lbs. of fascinating reading. It's yours for just $19.95.

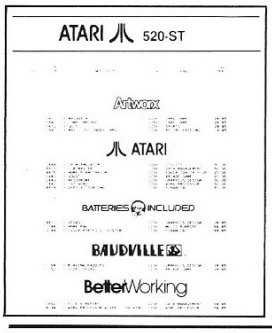
$19.95

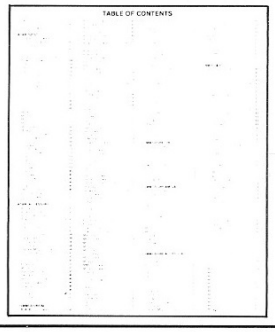
It's a dictionary of software, over 11,000 software titles, and filled with computers, accessories and other electronic products. It's yours for just $19.95.

Be your neighborhoods bulletin board for just $19.95. You will not only have what is available but a preview of the many products coming up that haven't reached your local store yet. In addition you'll have the retail price of each item. Illustrations show some of the information crammed pages. MAKE YOUR CHECK OUT TO CSS and mail to address listed at bottom. Add $2 for postage & handling. For quick service call our number and ask for Karen.







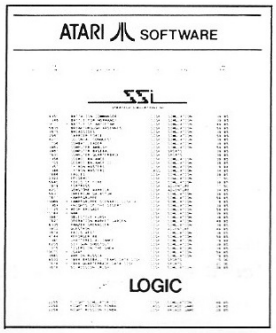


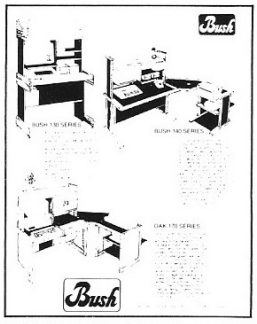
$19.95
Retail Price

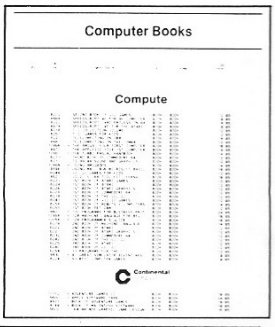
CSS
COMPUTER SOFTWARE & ACCESSORIES

LOCATOR SERVICE:

CSS is a distributor that sells nation wide to stores in your area. If you can't locate any of the items advertised in ST APPLICATIONS at your local dealer, call Karen at 1-800-422-4912 and we will tell you where you can buy it. Become the best informed person in your area.







CSS COMPUTER SOFTWARE SERVICE

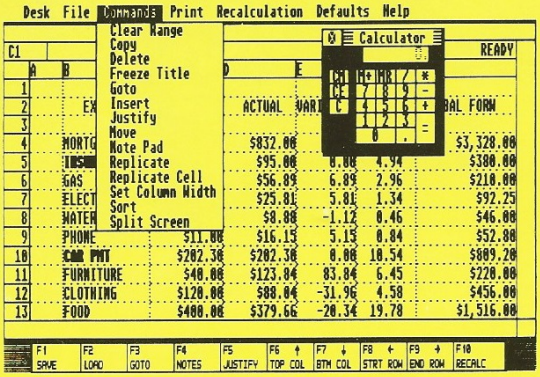
WE'VE GOT *THE MOVERS*®
495 A Busse Road • Elk Grove Village, IL 60007
(312) 439-4444 • Toll Free: 1-800-422-4912
In Illinois: 1-800-331-SOFT

EZ Calc™

Electronic Spreadsheet For Atari ST Computers

PUT EZ CALC™ TO WORK FOR YOU AND BENEFIT FROM THE RESULTS

EZ CALC™ is a fully implemented **GEM**™ based spreadsheet for home and business use. This is by far the most powerful spreadsheet available for the price. Better yet, all commands are mouse controlled for speed and ease of use. EZ CALC™ also uses **less memory** than other spreadsheets for the ST, leaving more room for your data and formulas. **If you've never used a spreadsheet before, you'll be amazed how easy** EZ CALC™ **is to learn and use. The experienced user will love the speed of a mouse controlled spreadsheet.**



ONLY $69.95

FEATURES

- 300 columns by 999 rows
- Extensive use of **GEM**™ windows
- All commands are under mouse control
- Built in 10 keypad calculator
- On-line help windows (No commands to memorize)
- Built in sort routine
- Developed exclusively for the Atari ST
- 10 macros controlled by the function keys
- Split-screen capabilities
- Note Pad

GEM is a Trademark of DIGITAL RESEARCH, INC.
EZ CALC is a Trademark of ROYAL SOFTWARE.

MOUSE CONTROL

Extensive use of the **GEM**™ windows make **EZ CALC**™ a fast, extremely easy-to- use spreadsheet. With over 50 commands available from the mouse, the ease of use is unsurpassed. Imagine being able to move or copy an entire column of figures with a simple mouse control.

CALCULATOR

EZ CALC™ includes an easy to use 10 key calculator that can be pulled down at anytime and operated either by mouse or keyboard. With the point of a mouse, the results of the calculation can then be transferred to the cell of your choice.

NOTE PAD

EZ CALC™ lets you attach a personal note of up to 4 lines to any cell. The cell is then highlighted to remind you there is a note attached. For example, you could attach a note to the insurance cell of your personal finance spreadsheet reminding you that the cell applied only to car and home insurance. The note pad can be pulled down at any time.

★ **FULL FEATURED DEMO OF EZ CALC AVAILABLE FOR $5. REFUNDABLE WITH PURCHASE.** ★

Help Calc ™

Only 24.95
For The Atari ST™

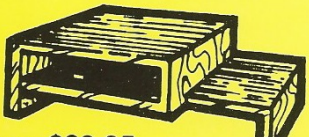
- 11 preprogrammed templates for use with **EZ CALC**™ or **VIP Professional**™
- **load-and-go** and these templates will take the work out of tedious spreadsheet setup.

Templates include:

- Check Register
- Depreciation schedules
- Investment Portfolio Analysis
- Name & Address directory
- Home Inventory
- Loan Amortization Schedules
- Personal Finance Statement
- and more

VIP PROFESSIONAL is a Trademark of VIP TECHNOLOGIES.

ST STAND



ONLY $39.95

+ $5.00 Min. Shipping & Handling

Custom made just for the ST, beautifully finished stand to hold your ST monitor, 2 disk drives, a modem, disk files, ETC...

DELUXE DUST COVERS

PROTECT YOUR EQUIPMENT

Deluxe Leather Grain

Made in Oregon

Custom fitted, attractive leather brown color:

7.95	• **KEY PAD**—ATARI CX85 • **PRINTER**—ATARI 1020 • **RECORDER**—ATARI 1010
8.95	• **COMPUTERS**—400/800 600/800/1200XL 65/130XE • **DISK DRIVES**—ASTRA 1620 ATARI 1050 ST DRIVES ST HARD DRIVE INDUS GT PERCOM 88S1/88SPD RANA 1000 TRAK • **PRINTERS**—ATARI 1025/1027 AXIOM SLP/GLP OKIMATE 10 • **RECORDER**—ATARI 410 • **MODEM**—ATARI 1030
9.95	• **COMPUTER**—ATARI 520/1040ST • **PRINTERS**—ATARI 825 AXIOM 550 CITOH/PROWRITER 8510 CPA 80/EP-150 EPSON MX/RX/LX/FX80 W. & W/O TRACTOR FEED GEMINI/STAR SG 10/X PANASONIC 1091/1092 RITEMAN II/PLUS
12.95	• **MONITORS**—B/W RGB MONITOR TEKNIKA MJ-10 • **PRINTERS**—EPSON FX 100/185

HELPMATE ™ Only $29.95

For The Atari ST™

HELPMATE™ **INCLUDES:**

- 10 Key calculator
- Appointment calendar with alarm functions
- Telephone/name index

All in one program

The best part is that HelpMate stays "hidden" in memory until needed, and then can be called up for use, even while another program is running. The pull down menus can be used with most ST programs or by themselves.

Coming Soon!!

INVENTORY MASTER ™

Only $99.95
For The Atari ST™

INVENTORY MASTER ™ is a powerful, Inventory control and Report generation program. It will do more than just keep track business inventory, such as: detailed report generation, fast and easy data retrieval, versatile data entry, takes the work out of decision making, plus much more.

COMPUTER PALACE

OPEN M-F. 9-6 Sat 10-4 (Pacific Time)

2160 W 11th Avenue, Eugene, Oregón 97402

USE YOUR CREDIT CARD & CALL

Toll Free 1-800-452-8013

★ ORDERS ONLY, PLEASE ★

There's never a penalty for using your credit card!

For Information, Call (503) 683-5361

Prices subject to change without notice.

SHIPPING INFO: Minimum $2.90 Ground $4.75 Air. Actual Cost depends on weight. Call (503) 683-5361 for information.
WARRANTY INFO: Everything that we sell is warranted by the manufacturer. If any item purchased from us fails to perform properly when you receive it, call us at (503) 683-5361 so that we can assist you. No returned merchandise accepted without authorization. Defective software will be replaced with another copy of the same program; otherwise, no software is returnable.

• 2 Day Air Shipping **AVAILABLE** •